

Devenez un développeur **Android**

+



=



A qui
appartient
mon code
source ?

Les nouveautés
de **Java 8**



Découvrez
Visual Studio 2013

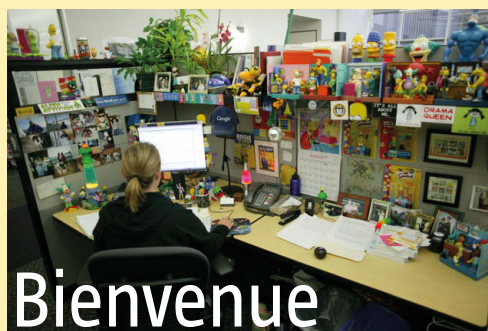
Windows Store
C++ 11
Windows Azure
Office 365 ...



La révolution **DevOps**

Le développement ne s'arrête jamais...

Bug : résoudre les
OutOfMemoryError en Java



Bienvenue
à la Silicon Valley



Printed in EU - Imprimé en UE - BELGIQUE 6,45 €
SUISSE 12 FS - LUXEMBOURG 6,45 € DOM Surf 6,90 €
Canada 8,95 \$ CAN - TOM 940 XPF - MAROC 50 DH

HTML5+jQuery

Toute App, Tout Navigateur, Toute Plateforme, Tout Appareil



IGNITEUITM
INFRAGISTICS JQUERY CONTROLS



Téléchargez Votre Version d'Evaluation Gratuite
www.infragistics.com/telechargez



INFRAGISTICS[®]
DESIGN / DEVELOP / EXPERIENCE

Infragistics Ventes France  0800 667 307 • Infragistics Ventes Europe +44 (0) 800 298 9055

Copyright 1996-2013 Infragistics, Inc. All rights reserved. Infragistics and Infragistics are registered trademarks of Infragistics, Inc. The Infragistics logo is a trademark of Infragistics, Inc.

BUZZWORD : PHP Week4

actualité6

RESSOURCES : La fiesta
du dev Mobile, DevFest, fOSSa10

matériel : Intel Galileo14

VEILLE TECHNO :
Une vie à la (Silicon) Valley18

**DÉVELOPPEUR
DU MOIS**22

OUTILS : TFS contre TFS,
Visual Studio 201324

Java 830

DOSSIER DEVOPS

.....34

JURIDIQUE : à qui appartient
le code que j'écris47

C# : maîtriser l'asynchronisme52

TESTING : Gatling55

azure : utiliser Media Services60

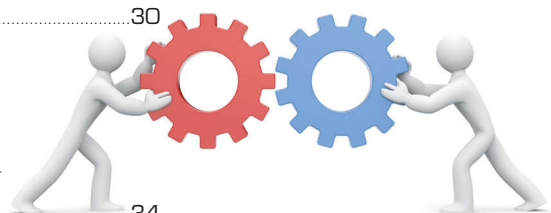
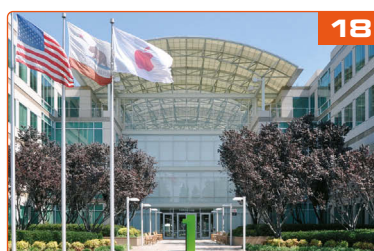
sÉCURITÉ : se protéger avec
Web Application Firewall62

Java : Guava by Google64

WEB : WebAPI et http68
Je démarre la programmation Android74

PHP : introduction à la sécurité
avec Symfony277

BUG : les outofmemory en Java81



édITO

Java 8, Visual Studio 2013, DevOps, OS X 10.9 : que de nouveautés !



Attention, embouteillages technologiques ! Les nouveautés se sont bousculées ces quatre dernières semaines. Côté Java, c'est la disponibilité des premières pré-versions développeurs (depuis mi-septembre). La version finale

devrait sortir d'ici mars 2014... Pour les développeurs Windows, quatre grosses nouveautés : Windows 8, Windows Phone 8 GDR3, Visual Studio 2013 et les nouvelles tablettes Surface. La GDR 3 supporte les écrans Full HD et les phablettes, avec la possibilité d'avoir plus de vignettes dynamiques, une optimisation pour les processeurs Snapdragon 800, et l'apparition du mode conduite. Le tout, avec une mise à jour du SDK...

Ce mois-ci, beaucoup de choses dans Programmez !. Vous allez découvrir en avant-première le tout nouveau microcontrôleur d'Intel, compatible Arduino, Galileo. Une petite merveille pour tous les bidouilleurs qui sera disponible fin novembre. Nous revenons également longuement sur le DevOps et en quoi la « philosophie » des déploiements continus révolutionne le développement ainsi que la manière de concevoir du code et des applications. Facebook, Twitter, Google, Netflix... ont adapté cette conception du code depuis le début.

<mode teasing>

Ne ratez pas le prochain numéro de Programmez ! Nous vous préparons beaucoup de surprises...

< fin du mode teasing>

François Tonic
Directeur de la publication
& rédacteur en chef
ftonic@programmez.com

A lire dans le prochain numéro

N°169-décembre 2013, à paraître le 30 novembre 2013

iOS 7

Les nouveautés, créer des applications, optimiser son code

Carrière

Osez le Cobol !

3D

Utiliser Unity3D avec PHP

Novembre : c'est la **PHP Week** !



Avec plus de 200 millions de sites web développés en PHP, le langage domine le Web depuis de nombreuses années ; il fallait bien consacrer une semaine complète pour satisfaire les aficionados de ce langage ! Deux événements majeurs annoncés à Paris : ZendCon Europe et le Forum PHP. Ne ratez pas la PHP Week !



Gain insights from peers, PHP luminaries, and thought-leaders

La PHP WEEK c'est une double dose de café : une semaine de folie pour et sur PHP, avec 2 événements majeurs :

- Tout d'abord la première édition de l'événement tournant ZendCon Europe dont la France sera la première étape en Europe, organisé par Zend Technologies.
- Dans un second temps, le Forum PHP, un événement annuel parisien organisé par l'AFUP (Association Française des Utilisateurs de PHP)

Ces 2 événements se dérouleront consécutivement du 18 au 22 novembre 2013.

Ce sera une grande première mondiale qui se tiendra en France, l'événement de la communauté se tenant la même semaine qu'une conférence éditeur renommée. Ces deux événements seront l'occasion pour les développeurs de passer une semaine complète autour de PHP.

> ZendCon Europe

Quelques semaines après la ZendCon US à Santa Clara, en Californie. Un nouvel événement tournant voit le jour, appelé « ZendCon Europe ». Il s'appuie sur le même principe que le PHP Tour en France, mais au niveau Européen. Pour la première édition 2013, ce rendez-vous s'arrêtera à Paris du 18 au 20 novembre.

Cet événement d'envergure européenne proposera un programme international de qualité avec la présence de conférenciers venus du monde entier. A noter que, bien qu'organisé par un éditeur, cet événement ne se limite pas aux produits de celui-ci, mais présente des sujets qui touchent l'ensemble des développeurs.

Un des éléments phares que vous ne pouvez

pas manquer, concerne la présence des deux co-architectes historiques de PHP et actuels CEO et CTO de Zend, Andi Gutmans et Zeev Suraski. Une keynote qui promet déjà d'être mémorable, les verra aux côtés de Fabien Potencier de Sensio Labs, créateur du Framework Symfony, où ils nous présenteront leur vision du futur du web.

Bien entendu, les auteurs de PHPUnit (Sébastien Bergmann), Xdebug (Derrick Rethans), seront parmi les dizaines de conférenciers internationaux qui effectueront le déplacement, ainsi que les différentes communautés PHP européennes qui auront un stand sur place.

En plus des 2 journées de conférences, ce premier rendez-vous sera, augmenté d'une première journée orientée tutoriaux faite pour vous aider à comprendre et utiliser certains outils et techniques innovants.

Par ailleurs, l'un de ces tutoriaux sera consacré à préparer la toute nouvelle certification PHP 5.5. Ainsi vous pourrez mettre

toutes les chances de votre côté pour obtenir la tant convoitée certification PHP lors de l'événement. Pendant les deux jours de conférence, une salle complète permettra en effet de passer les certifications PHP et Zend Framework. N'hésitez pas à consulter les différents créneaux programmés pour vous permettre de trouver le meilleur horaire par rapport à votre agenda. Ce tutoriel sera présenté par Gauthier Delamarre qui a déjà aidé des dizaines de développeurs à obtenir les certifications PHP et ZF.

La thématique principale concernera les bonnes pratiques autour du déploiement, de la maintenance du code à travers la documentation et les tests. Mais il sera aussi question de l'Apigility, une méthode déjà présentée à la ZendCon US, plus facile pour créer des API de haute qualité, et qui a déjà reçu un très bon accueil.

Soyez l'une des 450 personnes attendues pour cet événement, quelques surprises vous attendent sur place.

Pour connaître le programme complet, voici le lien direct :

<http://europe.zendcon.com/conference-sessions/>

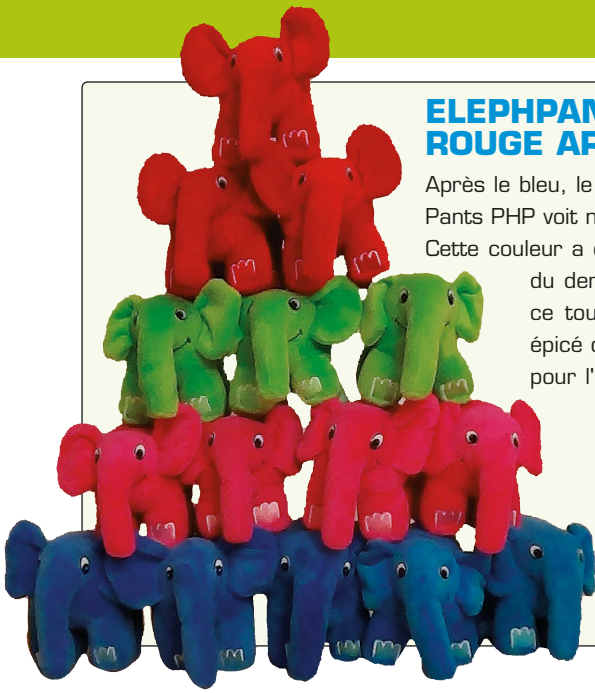
et le site officiel de l'événement

<http://europe.zendcon.com/>

> Forum PHP : la communauté française se réunit !

Pour la douzième année consécutive, l'AFUP (Association Française des Utilisateurs de





ELEPHPANT : ATTENTION, LE ROUGE ARRIVE !

Après le bleu, le rose et le vert, la famille des éléPH-Pants PHP voit naître une nouvelle couleur : le rouge. Cette couleur a été choisie par les développeurs lors du dernier sondage Zend Developer Pulse, ce tout nouvel elePHPant répond au nom épicé de Chili. C'est la septième portée, et, pour l'instant, la seule possibilité de l'adopter sera de venir à la 'PHP Week'. Nous rassurons bien entendu les lecteurs et les nombreuses familles déjà heureux parents d'un elePHPant que d'autres gestations sont en cours et seront disponibles très bientôt.

PHP) organise le Forum PHP les 21 et 22 novembre au Beffroi de Montrouge, à proximité de Paris. Cette édition sera axée autour de 4 thématiques, avec de nombreuses conférences en Français, qui correspondent au succès de PHP :

- Les Géants du web : PHP fait fonctionner les plus grands sites du web, c'est pourquoi les géants du web seront à l'honneur avec une problématique qu'ils rencontrent tous : comment scaler en PHP ?
- PHP est le langage de projets open-source leaders dans leur domaine. Ici les éditeurs de projets majeurs viendront présenter leurs solutions accompagnées d'un intégrant qui témoignera de son utilisation sur un cas concret.
- PHP est en constante évolution et l'innovation est aussi un axe fort de l'événement. Ainsi, vous découvrirez toutes les nouveautés récentes de la communauté PHP et du web.
- Les performances, l'architecture, l'agilité, le devops, seront abordés à travers différents retours d'expériences.

De nombreux acteurs de l'écosystème PHP feront le déplacement pour vous rencontrer. Nous pouvons signaler la présence de Julien Pauli, Release Master de la version PHP 5.5, qui en fera un tour d'horizon. Mais aussi la venue de Derick Rethans qui donnera une présentation sur OpenStreetMap. Ainsi que la présence d'Andi Gutmans, CEO de Zend Technologies.

Par ailleurs, la gestion de contenus sera aussi abordée à travers les différents CMS phares du moment. De nombreux retours d'expériences seront également assurés par des représentants de Google, M6 et Ouest-

France. Par ailleurs, de nombreuses cliniques sont en préparation pour animer ces 2 journées, grâce à la présence des sponsors Platinum et Gold. Celle de VA Consulting et Zend Technologies portera sur le Zend Framework 2 et celle de Théodo s'intitulera "Devops : accélérez vos déploiements pour une réactivité maximale". Ces cliniques aideront les équipes PHP à répondre à des problématiques précises sur ces thèmes. Pour connaître le programme complet, voici le lien direct : <http://www.afup.org/pages/forum-php2013/sessions.php> et le site officiel de l'événement <http://www.forumphp.org>

> Certification

Suite à la sortie de la version de PHP 5.5 en juin 2013 et à l'occasion du ZendCon US, une nouvelle version de la certification a été mise à jour, dont certaines questions porteront spécifiquement sur la version 5.5.

Le contenu a évolué par rapport à la précédente certification basée sur PHP 5.3, bien que de nombreux points restent toujours présents et inchangés. A noter, l'apparition des traits, l'évolution des Namespaces, de l'API de hashage ou de la méthode magique qui ont fait leur apparition dans cette nouvelle mouture du langage.

Certains termes ne seront pas identifiables directement car ils sont associés et couverts par le thème « SPL », comme les générateurs, arrivés dans PHP 5.5.

Pour rappel, l'examen a été conçu pour certifier de façon impartiale des aptitudes, des connaissances et des compétences particulières qui répondent à des besoins professionnels réels. Les certifications Zend fournissent aux employeurs un critère d'éva-

luation technique pour les employés ou futures recrues.

De plus, si vous souhaitez connaître les détails autour de la certification, vous pouvez vous rendre sur cette page. Vous y trouverez de la documentation et toutes les informations nécessaires pour devenir certifié et rejoindre cette grande famille qui vient de fêter son 10 000e membre.

<http://www.zend.com/fr/services/certification/>

> Le plus de la semaine PHP Week

La 'PHP Week' qui démontre que la France est pionnière sur ce domaine, se terminera par la publication des résultats du baromètre des salaires.

Ce baromètre a été construit d'après un sondage réalisé en partenariat entre l'AFUP et Agence-e, agence de recrutement, qui dévoilera ensemble les résultats autour de 4 axes pendant le Forum PHP :

- La rémunération
- Les fonctions PHP dans l'entreprise
- Les employeurs
- PHP et vous

Vous avez toujours la possibilité de participer au sondage, de façon anonyme pour mieux cerner ce secteur en constante évolution, et qui doit s'adapter aux nouvelles technologies web. Site : <http://www.afup.org/ask/barometre/2013.html>

> Conclusion

Les différents sondages et études publiés au fil des années montrent que PHP est très présent dans différents projets web, de petite, moyenne et grande taille. Participer à la 'PHP WEEK' vous permettra de comprendre pourquoi PHP a été choisi pour tous ces projets. Venez découvrir pourquoi il est de plus en plus utilisé dans des projets critiques par de grandes entreprises et pourquoi de plus petites structures en font le langage de référence pour l'ensemble de leur S.I.



Christophe Villeneuve
Consultant IT pour Neuros,
auteur du livre « PHP &
MySQL-MySQLi-PDO,
Construisez votre application
», aux Editions ENI. Rédacteur
pour WebRIVER, membre des
Teams DrupalFR, AFUP,
LeMug.fr, Drupalagora, PHPTV.



Christophe Chervy
Responsable marketing Europe
pour Zend Technologies et
membre du bureau de l'AFUP.

90 % des développeurs mobiles supportent plusieurs plate-formes

L'éditeur Appcelerator a dévoilé son étude sur les développeurs mobiles (Q2 2013 Mobile developer report). Cette étude porte sur 6 000 développeurs du monde entier en avril dernier. Premier constat : les développeurs mobiles sont de plus en plus en entreprise. Et les App Store d'entreprise connaissent une grosse croissance. Les tablettes continuent à attirer les développements (81 % contre 84 % pour le smartphone), par contre sur la partie Android, les développeurs considèrent les tablettes à 66 % seulement. La plate-forme iOS demeure la première destination, Android arrive juste derrière. Windows Phone connaît depuis quelques mois une croissance de l'intérêt des développeurs, un peu moins de 40 % ; à voir comment la plate-forme se comportera dans les prochains mois. Pour les tablettes Windows, la courbe fléchit légèrement et passe sous Windows Phone. Le développement mobile HTML 5 continue à être fortement prisé (+ 60 %).

Les autres plate-formes :

- BlackBerry : après une forte chute depuis janvier 2011, l'intérêt remonte un peu, environ 12 %
- Tizen : le système commence à apparaître (environ 5-6 %)
- Firefox OS semble prometteur, plus de 20 % d'intérêt

Le développeur mobile est un public à la recherche des nouveautés et des tendances. Ce n'est pas une surprise de voir Google Glass et la télévision connectée apparaître dans les designs et plate-formes à surveiller. Et surtout, 61 % des développeurs interrogés prédisent une poussée,

dans les 2 ans du MBaaS (Mobile Backend-as-a-service), c'est à dire l'usage de services Cloud par les applications mobiles (stockage, traitement déporté, etc.). Sans surprise non plus, 90 % des développeurs disent supporter et programmer sur plusieurs plate-formes. Aujourd'hui, une application ne peut plus être disponible sur un seul système. Sur Windows Phone et les tablettes Windows 8 / Windows RT, l'équation à résoudre est simple : pas de développeurs = pas d'applications = faibles ventes de terminaux. Mais on peut inverser l'argument : faible vente = peu d'intérêt des développeurs = peu d'applications. Cependant, on voit un réel intérêt de la plate-forme avec

des ventes en hausse et des parts de marché à 10 % dans plusieurs pays.

Clairement, Microsoft est le 3e système mobile. Le recruteur américain CyberCoders a mené une enquête sur les salaires des développeurs Android et iOS.

Et là, petite surprise, un développeur senior Android peut espérer un salaire de 131 000 \$ contre 121 000 pour son équivalent iOS ! Par contre, pour un développeur « normal », le salaire est équivalent entre les deux plate-formes, 107 000 \$.

Rapport :

<http://www.appcelerator.com/thinkmobile/surveys/>



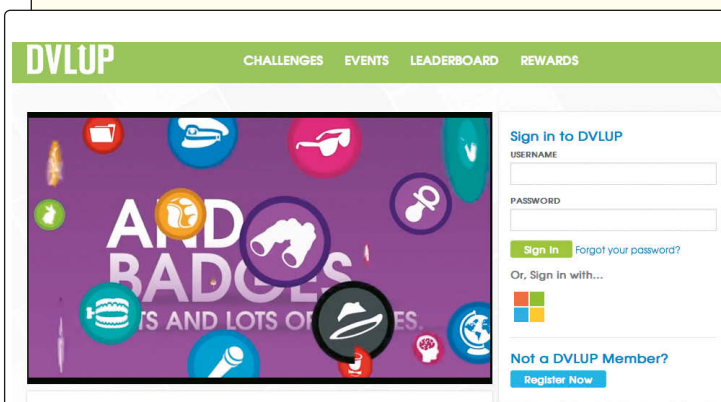
Nokia veut aider le développeur avec DVLUP

Pour promouvoir la plate-forme Windows Phone et les terminaux Lumia, le constructeur propose un challenge : DVLUP. Le but est très simple : développer une application. Chaque challenge est basé sur une idée. La réussite va dépendre du respect des règles, de l'utilisation des technologies et des fonctionnalités décrites, ainsi que du résultat final. Chaque challenge réussi donne des points au développeur, un peu comme les Miles des compagnies aériennes. En cumulant des points, le développeur peut acheter une campagne de promotion de son application, du matériel, des goodies, etc. Pour Nokia, il s'agit de valoriser la plate-forme, de promouvoir certaines fonctions

(ex. : Imaging SDK, NFC) et de développer l'écosystème et l'intérêt des développeurs.

Chaque challenge dure en général 4 semaines, le développeur a le temps de concevoir et de développer. DVLUP est ouvert dans 24 pays dont la France.

Site : <http://www.dvlup.com>



Xbox One : OpenGL et Mantle out !

Microsoft a beaucoup travaillé sur ses API 3D / 2D : DirectX, Direct3D, etc. Avec Windows 8.1 et surtout la Xbox One, Direct3D passe une nouvelle étape avec Direct3D 11.x. La console supporte la version 11.2. Cela signifie aussi le non support des bibliothèques OpenGL et Mantle. Mantle est une bibliothèque 3D d'AMD pour ses GPU.

Microsoft met en avant le même modèle de développement entre Windows et Xbox avec DirectX3D qui bénéficie d'améliorations dans Visual Studio pour le debug et l'analyse.

EN ROUTE POUR LES NOUVEAUX DOMAINES

Vous pouvez maintenant choisir parmi **plus de 700 nouvelles extensions de domaines** pour créer l'adresse originale et facile à retenir qui vous correspond le mieux, comme par exemple **monrestaurant.paris**, **magalerie.art** ou **legrand.shop**. Pré-réservez aussi de nouvelles extensions pour vos domaines existants. Il sera d'autant plus facile de vous trouver !

Avec environ 20 millions de domaines enregistrés, 1&1 est leader du marché de l'enregistrement de noms de domaine en Europe. Grâce à la **fonction de redirection intégrée**, les domaines enregistrés auprès de 1&1 peuvent rapidement et facilement être redirigés vers n'importe quel site Web, quel que soit votre hébergeur.

Plus d'informations sur **1and1.fr**

**NOUVEAU !
PRÉ-RÉSERVEZ
SANS FRAIS
ET SANS ENGAGEMENT***



DOMAINES | MAIL | HÉBERGEMENT | E-COMMERCE | SERVEURS

1and1.fr

* La phase de pré-réservation est gratuite et sans engagement. La disponibilité et le prix des domaines dépendent du registre compétent : 1&1 s'engage à communiquer ces informations dès qu'elles seront disponibles. L'enregistrement du domaine est soumis aux conditions générales de vente de 1&1 et à celles du registre compétent. 1&1 ne peut garantir l'attribution finale des domaines pré-réservés.

Mozilla Game Jam Paris 2013 : le show 100 % gaming

Pour sa première édition à Paris, le Mozilla Game Jam s'est déroulé sur deux jours complets dans les locaux de la fondation Mozilla. Plus de 50 passionnés s'y sont pressés. Programmez ! y était pour vous !

Une Game Jam est un hackathon de création de jeux vidéos, dont les réalisations doivent fonctionner sur Desktop, smartphones, tablettes... et bien entendu sur Firefox OS. Le langage de réalisation doit être ouvert, et s'appuyer sur les technologies du web : HTML /HTML5, CSS, javascript... Les participants étaient de différents niveaux : passionnés avec une soif d'apprendre et avec des connaissances plus ou moins poussées au niveau développement, graphisme ou musique... objectif : se faire plaisir ! Cet événement n'imposait pas de thème. Il permettait de se rencontrer, proposer et présenter des idées, des concepts et de les réaliser...



Fig.1



Fig.2



Fig.4

> Déroulement

Le déroulement d'une 'Game Jam' commence par la présentation de son idée de jeu à l'ensemble des personnes inscrites. Ces personnes choisiront le jeu qui les intéresse, et les équipes se forment. La sélection a permis de voir 5 équipes dont l'objectif était de présenter un jeu fonctionnel et jouable à la fin de l'événement. Après la composition des équipes, chacune d'elle définit les rôles de chacun pour mener à bien le projet dans les délais impartis. Enfin, 48 heures après, les réalisations étaient présentées par les équipes pour le plaisir de tous [Fig.1].

> Les réalisations Spaceship Mercenaries

par les Power Rangers

Spaceship Mercenaries consiste à détruire le vaisseau de l'adversaire avec les pouvoirs de son propre vaisseau, sur un même terminal en utilisant le multitouch. L'énergie représente la vie de chaque joueur, mais aussi sa puissance. Pour lancer une action parmi les 3 disponibles vous oblige à utiliser les actions avec parcimonie et au bon moment, en fonction des actions de l'adversaire. Ce jeu s'appuie sur le principe d'un jeu de duels tactiques [Fig.2]. Jeu disponible : <http://jmpp.fr/games/mozgamejam2013/>

Game.random(), par les Semi-croustillants

Il s'agit de plusieurs mini jeux dans un seul jeu, chacun étant choisi au hasard par l'ordinateur. Un bouton 'changement' permet encore de changer de jeu au hasard. Vous trouverez ainsi :

- Redessiner la forme graphique

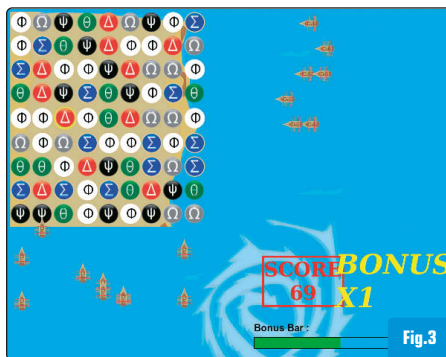


Fig.3

- Eviter les astéroïdes
- Tuer des chats qui volent
- Eviter les obstacles
- Faire atterrir un vaisseau
- Un snake
- Cliquer sur un objet

Le but de ce jeu est de vous permettre de jouer quelques minutes lors d'une petite pause.

Jouer en ligne <http://www.thibault-eynard.com/divers/Game.random%28%29/>

Télécharger le jeu <http://www.thibault-eynard.com/divers/Game.random%28%29.rar>

Syracuse, par les Deadly Worms

Il s'agit d'un jeu exclusivement sur PC. Il s'inspire du genre match 3 / bejeweled qui tire partie des coordonnées des formes afin d'introduire une nouvelle dimension au gameplay. Une grille représente l'île de Syracuse, dont le joueur doit combattre une invasion de bateaux romains. Pour les détruire, vous devez aligner 3 formes identiques pour stopper les bateaux qui avancent vers l'île [Fig.3].

Jeu en ligne : <http://games.lqbs.fr/syracuse/>

Télécharger l'archive : <http://games.lqbs.fr/syracuse/syracuse.zip>

Janken, par les BN (Bestial Nigga)

Il s'agit d'un jeu de combat inspiré du jeu classique pierre-feuille-ciseaux, avec la particularité, lorsque vous gagnez, d'attaquer votre adversaire. A chaque attaque, la barre d'énergie du joueur augmente, jusqu'à déclencher une attaque spéciale plus puissante. De plus, vous pouvez choisir votre personnage parmi les 3 disponibles. Chacun d'eux possède des statistiques différentes (énergie, attaque spéciale, etc...). Jeu en ligne : <http://alvin-ntumba.com/janken/>

TipTape LapTape, par les Gwakamole

Il s'agit d'un jeu inversé dont vous êtes la taupe et non le maillet, et inspiré du jeu Whac A Mole. Le but est de survivre le plus longtemps possible aux attaques. Il est possible de faire sortir plusieurs taupes en même temps pour augmenter le nombre de points. Par ailleurs, suivant la durée de pression du bouton, votre taupe pourra rester plus longtemps sortie. Ainsi, si vous vous faites toucher, vous perdez des points [Fig.4]. Jeu en ligne : <http://gwhackamole.github.io/Game/> .Télécharger l'archive : <https://github.com/gwhackamole/Game>

> Conclusion

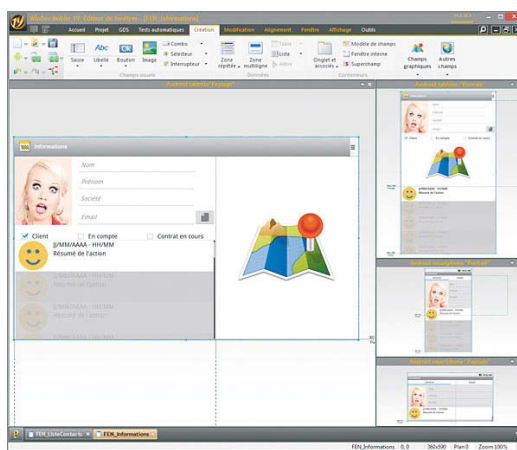
Les organisateurs ont promis une nouvelle édition dans les prochains mois. Il s'agit d'un événement ouvert et même si vous ne connaissez pas à 100 % les langages du Web, c'est un bon moyen de venir apprendre en vous amusant. Ce type d'événement peut être organisé dans différentes villes, comme la prochaine édition à Lyon. Site : <http://gamedevparty.fr/gdp-jam-5-reservez-votre-week-end/>

Christophe Villeneuve.
Consultant IT pour Neuros.

WinDev 19 : widgets, Android / iOS, 64 bits, JSON, JQuery,

L'éditeur PC Soft a annoncé la version 19 de son atelier intégré de conception logicielle : WinDev. Le Cloud Computing est un axe prioritaire pour PC Soft. La v19 permet de stocker et de partager les projets en ligne (via le service PCSCloud). On dispose ainsi des outils de contrôle de qualité et de suivi projet. Plus besoin d'installer l'environnement : vous pouvez utiliser et accéder aux projets depuis un poste inconnu... Autre nouveauté appréciable, les éditeurs de WinDev, WebDev et WinDev Mobile sont désormais 32 et 64 bits, permettant d'utiliser plus de ressources. Le tableau de bord projet permet désormais de s'afficher sous forme de widgets. Cela permettra une plus grande personnalisation de ce module. Et selon le projet, vous pourrez utiliser les widgets nécessaires.

Sur la partie code, l'éditeur améliore la création et l'accès aux propriétés des classes, supporte les accès SharePoint et la complétion sur les sources de données... En mode debug, on peut définir à la volée l'instruction devant être exécutée et même revenir en



arrière (en jouant avec la flèche jaune). Le debug permet aussi de voir le contenu des requêtes.

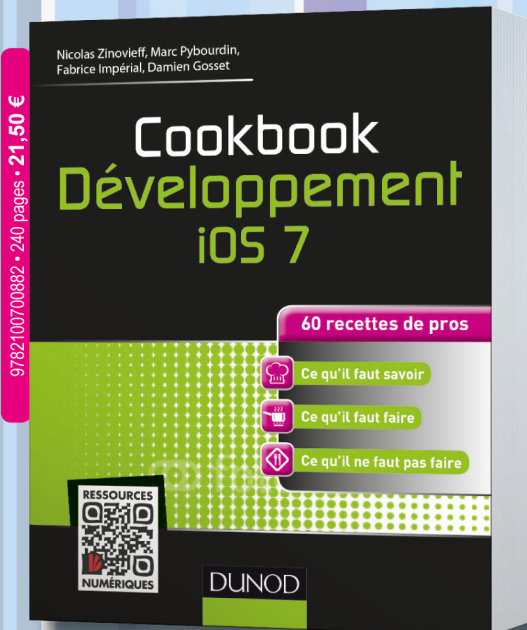
La partie base de données évolue aussi. Désormais, HyperFileSQL devient tout simplement, HFSQL. Elle est maintenant "ACID" (Atomicité, Cohérence, Isolation, Durabilité) et propose plusieurs niveaux d'isolation des transactions. D'autre part, les équipes ont travaillé à optimiser la mémoire de HFSQL, notamment avec le passage au 64 bits (ges-

tion des caches). De nouvelles commandes SQL font aussi leur apparition ainsi que des accès aux bases Informix, DB2... La v19 supporte même le NoSQL (MongoDB).

Sur la partie mobile, PC Soft a amélioré le support Android / iOS, en particulier pour aider le développeur à préparer les interfaces en mode portrait et paysage et à faciliter la création mobile multiplateforme (notamment sur des fenêtres répondant aux contraintes de chaque plateforme). Les données mobiles peuvent être répliquées avec le serveur HFSQL. PC Soft annonce aussi le support de iOS 7, des derniers mobiles iPhone, du InApp, de la notation. La partie Android bénéficie aussi de nombreuses évolutions : évaluation, réplication, push, nouveaux graphes, gestion connexion / déconnexion des réseaux... Les précommandes peuvent se faire dès maintenant. L'éditeur annonce que certaines fonctions ne seront pas disponibles immédiatement. Les éditeurs 64 bits sont annoncés pour février 2014.

Site : <http://www.pcsoft.fr>

Maîtrisez les nouvelles technologies



60 recettes pratiques afin d'effectuer les tâches les plus fréquemment rencontrées pour **créer une application avec iOS 7**

Ce guide pratique est le compagnon idéal pour exploiter au mieux les multiples possibilités du Samsung Galaxy S4



novembre

DevFest Nantes 2013 : 8 novembre à Nantes

Les DevFest sont des événements communautaires de grande envergure à travers le monde, suivant la tradition des Google Developer Days, et organisés par des groupes d'utilisateurs locaux. Ils constituent une occasion unique de partager et d'échanger autour des technologies du Web et du Cloud ! Et plus particulièrement sur Android, Dart, AppEngine, Angular JS, Chrome, HTML5, WebGL, Google Drive, Go, GWT... Cette journée est organisée par Le GDG Nantes. Site : <http://devfest.gdg Nantes.com>



Green Code Lab Challenge : du 27 au 29 novembre



Proposé par le Green Code Lab et co-organisé par l'ESAIP Angers, le Green Lab Center, Open Odyssey et l'EPIS Nantes, l'objectif principal du Challenge est de démontrer l'importance de l'éco-conception logicielle dans le monde de l'informatique, de l'enseignement supérieur et de la recherche. Dans un contexte où l'usage des technologies de l'information explose, il faut impérativement minimiser le fort impact de ces solutions sur l'environnement. L'éco-conception prend peu à peu son envol auprès des développeurs et des entreprises.

Site : <http://greencodelab-challenge.org>

Open du web #4 : le 28 novembre à Paris

C'est l'événement web et du recrutement web de novembre. L'Open du web est un concours et un speed-dating autour des métiers du web. Mission : des centaines de développeurs, designers s'affrontent pour réaliser un projet et démontrer leur maîtrise ! Durant l'événement, des conférences et ateliers se succéderont. Cloudmagazine est partenaire et ne ratez pas la conférence de François Tonic sur "comment le cloud modifie le web". Site : <http://www.open-du-web.fr>



agenda

NOVEMBRE

TDF Tech 2013 : PC Soft organise comme chaque année son tour de France technique. Occasion de voir et de toucher les versions 19 des outils de l'éditeur. Quelques dates : Montpellier (5 nov.), Toulouse (12 nov.), Bordeaux (13 nov.), Nantes (14 nov.), Paris (19 nov.). Tous les détails sur : <http://www.pcsoft.fr/pcsoft/tdfcom/2013/index.html>

DECEMBRE

Botconf'13 : Entièrement dédiée à la lutte contre les Botnets, elle est organisée par un comité scientifique indépendant et international. En Anglais. Lieu : Nantes, les 5 et 6 décembre. Site : <https://www.botconf.eu>

La fiesta du dev mobile : les 23 et 24 novembre à Paris

Vous avez la passion du développement mobile ? Vous voulez en savoir plus ? La fiesta du dev mobile est faite pour vous ! Durant 30 heures, dans toute la France, vous allez pouvoir voir, rencontrer les communautés et les développeurs, et toucher les plateformes et technologies. Et en plus, vous pourrez vous frotter aux développeurs dans un championnat de France du développement mobile, avec de nombreux lots à gagner ! Plusieurs catégories seront proposées : Android, iOS, Windows Phone, design et prix spécial du jury. Pour les développeurs, des modules de difficultés variées seront proposés pour les 3 grandes plateformes. A chaque module seront affectés des points en fonction de sa difficulté. Vous pouvez concourir seul, à 2 ou à 3. Le tarif est dégressif si vous êtes en équipe, mais les points ne sont pas calculés de la même manière... Pour les designers, le principe consistera à proposer le plus beau design avec la meilleure ergonomie, en s'appuyant uniquement sur les wireframes présentés. Les designers ne peuvent pas concourir en équipe. Le jury se compose de personnalités du monde numérique, du web, du mobile. "Nous sommes tous aujourd'hui intimement liés à nos smartphones. Mais au-delà du matériel et des grands constructeurs, ce sont plus de 2 millions d'applications mobiles que nous utilisons et qui ont été créées par des développeurs talentueux." explique Gilles Feingold, Président de l'AFPDM. Site : www.lafiestdudev.mobi



décembre

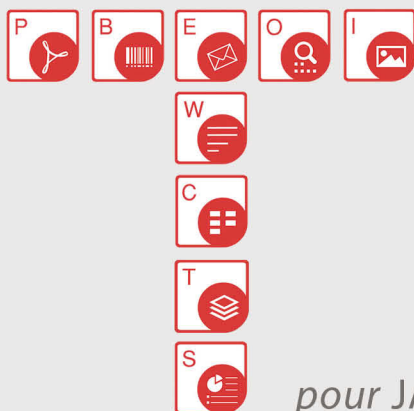
APIDays : 4 & 5 décembre à Paris

Le monde des API évolue tout le temps. Les API sont désormais au cœur des entreprises et dans la vie quotidienne. APIDays revient en France avec plus de 50 speakers internationaux pour l'édition globale de la conférence dédiée aux APIs. Cette année, deux axes seront mis en avant : l'expérience utilisateur et les API dans les processus internes des entreprises. La dimension développeur n'est pas pour autant oubliée. De nombreuses conférences techniques et stratégiques auront lieu durant les deux jours.

Site : <http://paris.apidays.io/2013/>



ASPOSE.TOTAL POUR JAVA



pour JAVA

Convertir, Imprimer, Créer
Combiner & Modifier

DOC PPT
PDF MSG
XLS VSD

formats d'images et beaucoup PLUS

ASPOSE.TOTAL POUR PRODUITS JAVA

PROGRAMMATION RAPIDE ET SIMPLE

Aspose.Words pour Java: Créer, modifier et convertir les documents Word et les autres fichiers de traitement de texte. Contrôle du contenu, effectuer le publipostage et convertir des documents aux formats de fichier natifs.

Aspose.Pdf pour Java: Créer, modifier et convertir les fichiers Adobe Acrobat PDF et les pages d'image et d'autres formats de fichiers. Travailler avec des formulaires, signatures et les documents sécurisés.

Aspose.Slides pour Java: Créer, modifier et convertir les présentations Microsoft PowerPoint. Créer les diapositives de scratch, enregistrer les diapositives, recourir à des sources de données externes et de protéger les présentations.

Aspose.BarCode pour Java: Lire et générer les codes à barres dans une gamme populaire 1D & 2D.

Aspose.Email pour Java: Travailler avec Microsoft Outlook et les fichiers d'archivage, de réception et d'envoi. Synchroniser les e-mails.

Aspose.OCR pour Java: Exécuter la reconnaissance de caractères dans différentes langues. Travailler avec un OMR pour construire des applications puissantes et efficaces.

Aspose.Imaging pour Java: Créer, modifier et convertir des images rastées et vectorisées. Travailler avec Adobe PhotoShop au format PSD pour dessiner des images et les préparer pour un traitement externe.

Ventes Europe:
+44 141 416 1112
sales.europe@aspose.com

 **ASPOSE**
Votre Expert des formats de fichiers

SCANNEZ POUR
REMISE DE -20%



Deux anciens de l'Epita au cœur de Google et Facebook

Dans les locaux de l'Epita, le 20 septembre dernier, Christophe Chedeau (promotion 2012) et Pierre Bourdon (promotion 2013) présentaient leurs parcours, de leurs études à l'EPITA jusqu'à leurs postes actuels : ingénieur Front-end chez Facebook pour le premier et site reliability engineer (SRE) chez Google pour le second. Une belle aventure.



« Je suis entré chez Google par le biais des multiples contacts que j'ai pu me faire durant mon cursus, commence Pierre Bourdon. Quand je suis entré à l'EPITA, je me suis investi dans des associations - Prologin, entre autres - et des laboratoires comme le LSE, tout en continuant de m'impliquer dans diverses communautés sur Internet. Dans ces différentes structures, j'ai pu faire mes preuves, j'ai été amené à prendre la parole en public plusieurs fois et j'ai mené des projets complémentaires à mes études. Tous ces éléments, en plus des multiples contacts que j'ai pu nouer, ont forcément joué en ma faveur lorsque j'ai répondu à l'offre de stage proposée par Google Zurich. »

« Google fait travailler ses ingénieurs dans un esprit très américain, précise-t-il. Tout est fait pour que les ingénieurs se sentent bien : des activités ludiques sont mises à disposition, la nourriture est offerte, le flux de travail est libre... L'idée est que nous soyons à l'aise pour travailler efficacement. »

« Venir au travail est amusant, assure Christophe. Les développeurs et ingénieurs informatiques sont la clé de voûte de l'entreprise. Tout est fait pour qu'ils s'y sentent à leur place. Pendant mes études, j'alimentais régulièrement un

blog où je présentais mes travaux et projets développés à l'EPITA. La technicité de la mise en application des enseignements de l'école a intéressé un des directeurs en ingénierie de Facebook qui me lisait. Il m'a alors contacté. Après plusieurs échanges téléphoniques pour évaluer ma motivation et ma culture informatique, ils m'ont fait venir en Californie pour passer différents entretiens. Une semaine après, j'étais pris. »

« A Facebook, je constate que les compétences que j'ai acquises à l'EPITA valent, voire dépassent, les cours très théoriques de certains de mes collègues sortant de Stanford ou du MIT. Le fait de mettre les étudiants dans le projet les prépare directement au monde du travail. » résume Christophe.

Un conseil aux étudiants ?

« Je pense qu'il faut profiter de ses années d'études pour multiplier les expériences, répond Pierre. Le monde associatif est un microcosme où tout le monde se connaît très rapidement. Idem pour les laboratoires. Sur les dix Anciens Epitéens travaillant à Google Zurich, trois étaient actifs au sein de Prologin, par exemple. Il faut profiter de ces réseaux ».

Thomas Hajdukowicz
journaliste web chez EPITA

novembre

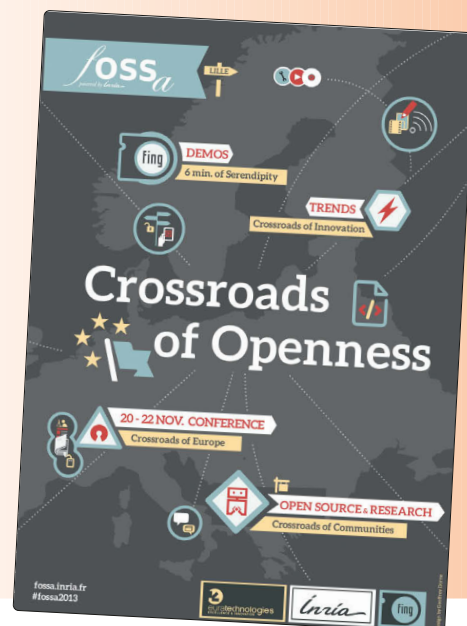
OW2Con'13, 12-14 novembre à Issy-les-Moulineaux

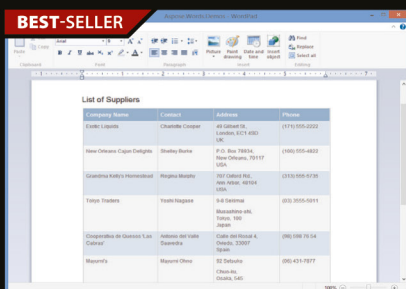


OW2 organise sa grande conférence annuelle. Elle aura lieu à Issy-les-Moulineaux. OW2 héberge et incube une centaine de projets. Cette 5e édition réunira l'industrie, les développeurs, les architectures, les experts des différents projets. La 1ère journée sera dédiée ateliers. Cette année, plusieurs thèmes seront incontournables : big data, BI, cloud computing, open source, la qualité logicielle. Site : <http://www.ow2.org/view/OW2con-2013/>

fOSSa, 20-22 novembre à Lille

La 5e édition de fOSSa, la Free Open Source Software for Academia se tiendra à Euratech-nologie (Lille) du 20 au 22 novembre. L'événement est organisé par l'INRIA, Euratechnologies et la Fing. fOSSa, ce sont 3 jours intenses de conférences, débats, ateliers, rencontres participatives avec des experts internationaux, qui replacent l'open source à la croisée de l'industrie et de la recherche, en mode bureau des tendances internationales. C'est aussi l'occasion de voir l'impact des technologies sur la société et toutes les tendances de l'open hardware, l'open network, etc. Plusieurs parcours thématiques seront proposés, sans oublier les rassemblements communautaires. Site : <https://fossa.inria.fr/program/>



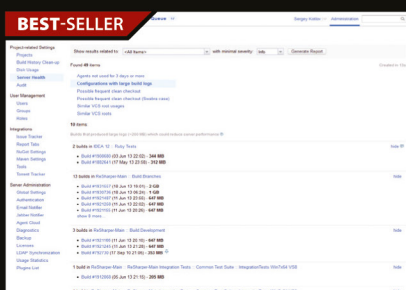


Aspose.Words for .NET à partir de € 731



Lisez, modifiez et écrivez des documents Word sans Microsoft Word.

- Création de documents, manipulation du contenu/formatage, puissante capacité de fusion de courrier et exportation en DOC/HTML
- Accès détaillé à tous les éléments d'un document par programmation
- Support les formats de fichiers: DOC, DOCX, WordprocessingML, RTF, HTML, OOXML, OpenDocument, PDF, XPS, EMF et EPUB

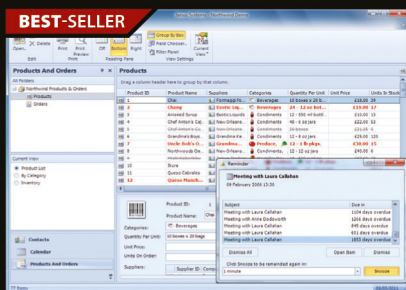


TeamCity à partir de € 1 763



Serveur d'intégration continue prêt à l'emploi, extensible et conçu pour les développeurs.

- Configuration rapide et intuitive des projets de builds
- Gestion intelligente de serveurs (usage des disques/état serveurs)
- Analyses et couvertures de code intégrées
- Extensibilité simple via REST API, messages des scripts de builds et API ouverte
- Support intégré pour une vaste gamme de technologies incluant Java, .NET, Ruby, Objective-C et Android

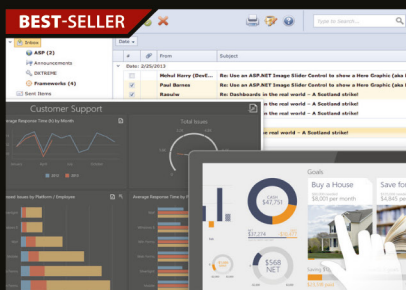


Janus WinForms Controls Suite V4.0 à partir de € 678



Ajoutez des interfaces de style Outlook à vos applications .NET.

- Vues ruban, grille, calendrier, et barres chronologique/raccourcis
- Nouveau – Style visuel Office 2010 pour tous les contrôles
- Nouveau – Support des profils client Visual Studio 2010 et .NET Framework
- Janus Ribbon ajoute Backstage Menus et la fonctionnalité onglet comme dans Office 2010
- Prend désormais en charge la sélection de cellules multiples



DevExpress Universal Suite à partir de € 1 611



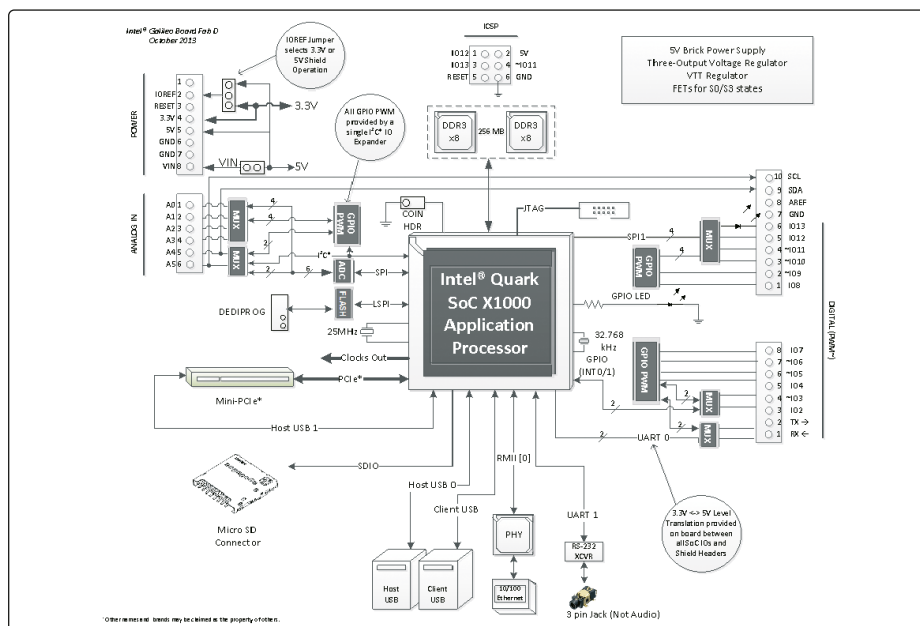
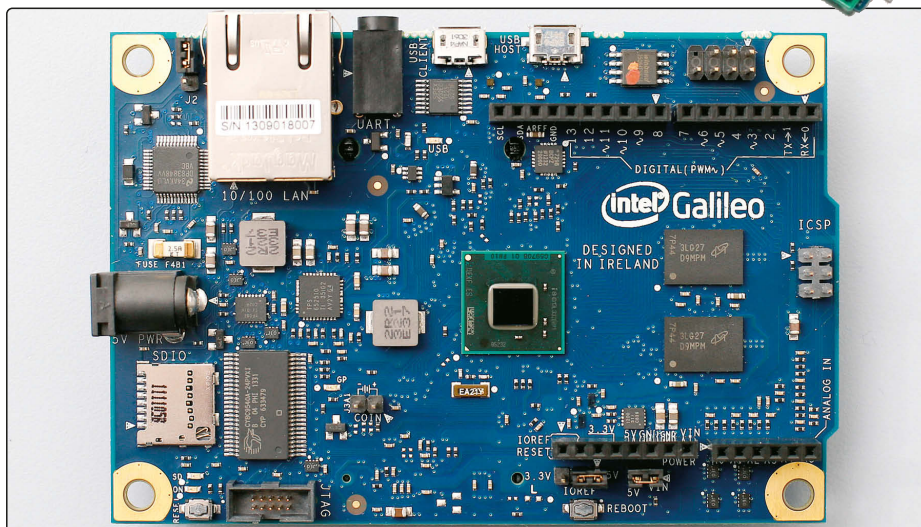
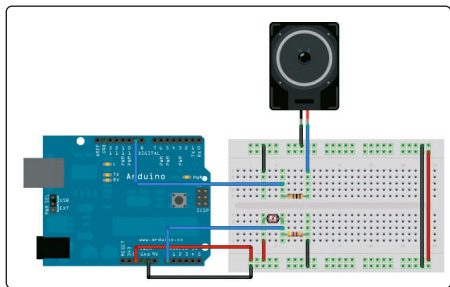
400+ outils et contrôles WinForms, ASP.NET, WPF, Silverlight et Windows 8.

- Exploitez votre base de codes pour développer des applications tactiles multiplateformes
- Inclut un tableau de bord de visualisation interplateformes et un serveur de rapports
- Nouveaux contrôles DevExpress incluant Tableur et Carte
- Codage, débogage et refactorisation avec CodeRush pour Visual Studio
- Inclut la nouvelle galerie unifiée de modèles applicatifs DevExpress

> Un microcontrôleur propre et complet !

Les possibilités sont infinies !

Galileo étant compatible Arduino, Intel propose par défaut l'environnement de développement Arduino (disponible sur OS X, Linux, Windows). Pour le langage, vous aurez l'embarras du choix tant qu'il supporte l'extension .586 pour processeurs x86. Le firmware est open source (et en C). Les compilateurs GCC et ICC sont supportés. Il est possible de déployer Linux. Pour les outils de développement, la communauté et Intel fournissent tout le nécessaire. Attention : mettre à jour le logiciel interne avant la première utilisation.



Seule, la carte ne sert pas à grand chose. Son intérêt est de l'interfacer avec d'autres objets, des circuits et de les piloter, les contrôler. Pour se convaincre des possibilités, jetez un œil sur les exemples officiels d'Arduino : <http://arduino.cc/en/Tutorial/HomePage> Et surtout, pour compléter la carte, vous pouvez acheter et utiliser les shields Arduino, ce sont des extensions matérielles : LED, modules entiers, GSM, relais, wifi, camera,

> Combien coûte le kit complet ?

Site : <http://www.intel.com/support/galileo/index.htm>

DevOps d'entreprise



- **Planifiez et contrôlez vos trains de release**
- **Automatisez vos livraisons d'applications**
- **Provisionnez les infrastructures, les données et les applications**
- **Réduisez les erreurs de mise en production**
- **Améliorez la communication entre les équipes**

Les délires **geek** pour la maison !



» **Art Lebedev**, studio de création et de design russe, a inventé la nouvelle chaîne de sécurité pour les portes d'entrée : defendius ! Pourquoi simplement glisser la fixation de la chaînette sur un simple rail ? Aucun intérêt. Les créateurs ont fait mieux : un labyrinthe ! Par contre, prévoyez un joli panneau d'avertissement : « sonnez puis attendez quelques minutes »... Il est possible de le trouver sur des sites comme think-geek.com

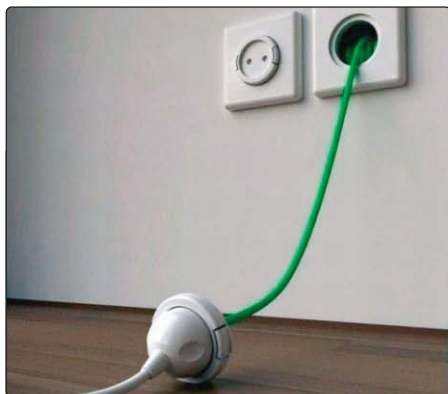
» PNY : cartes GeForce overclockées

PNY propose les cartes GeForce GTX 770 et 780 en versions overclockées. Les cartes gardent les caractéristiques constructeur mais PNY optimise les fréquences et rajoute un système de refroidissement dimensionné aux nouvelles contraintes. Ces cartes supportent aussi les fonctions TXAA, Adaptive Vertical Sync... Les GeForce GTX 770 et GTX 780 sont livrées avec une garantie de 3 ans et sont disponibles

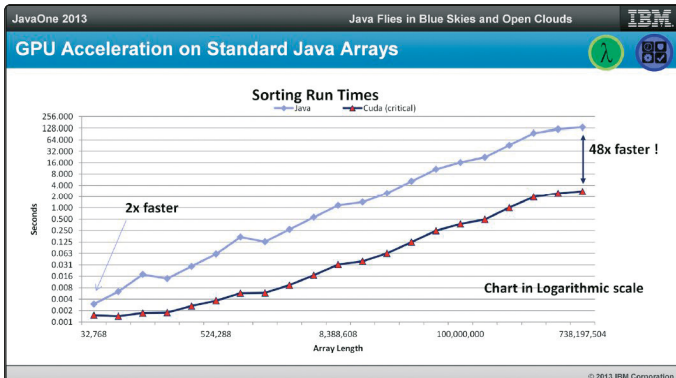
auprès du réseau habituel de détaillants de PNY pour 399 € (770 OC) et 629 € (780 OC).

» HP, nouveaux ultrabook

HP a dévoilé début octobre une nouvelle gamme d'ultrabook pour usage professionnel : HP EliteBook et ProBook. La gamme EliteBook annonce des modèles plus légers, une autonomie plus grande. Les modèles ont été soumis à des tests de solidité et de robustesse (MIL-STD 810G). La gamme ProBook offre une plus grande personnalisation de la configuration (notamment la 4G). Les écrans sont tous Full HD ou HD+. Les tarifs débutent à 579 € H.T.



» **La prise électrique avec enrouleur incorporé !** voilà un super concept que l'on aimerait avoir chez soi. Qui n'a jamais été embêté par une prise électrique ? L'idée est simple : avoir une rallonge intégrée à la prise (derrière le boîtier). Inconvénient : prévoir un grand trou pour poser l'enrouleur, en croisant les doigts pour qu'il ne se bloque pas...



» La GPU à la rescousse de Java ?

Durant la dernière conférence JavaOne, John Duimovich (IBM, chargé de Java) a démontré l'accélération Java grâce aux GPU. Il s'agit, comme toujours dans ce type d'accélération matérielle, d'utiliser les ressources de la carte graphique pour certaines opérations. Celles-ci peuvent atteindre un facteur 10 sur certaines tâches comme les calculs complexes, domaine où les GPU sont très bons. Java est souvent critiqué par sa lenteur et cette solution pourrait améliorer les choses. La piste explorée est d'utiliser dans Java 8 les avantages et traitements CUDA (à condition d'utiliser les bibliothèques CUDA dans son code Java).

» Nouvelles cartes Radeon série R

AMD a dévoilé six nouvelles cartes Radeon de la gamme R. Deux fonctions sont mises en avant pour le fondeur : Mantle et TrueAudio. TrueAudio doit permettre un réalisme audio de très haut niveau pour les jeux et donc pour l'expé-

rience gamer. Mantle est la technologie 3D optimisée et adaptée aux GPU AMD. Le constructeur annonce aussi des performances toujours plus hautes pour le C++, OpenCL et C++ AMP. Pour en savoir plus : <http://www.amd.com/us/products/technologies/gcn/Pages/gcn-architecture.aspx>

» Une imprimante 3D à - 800 €

PEARL Diffusion annonce une impression 3D à un tarif très agressif : - 800 € ! Il s'agit du modèle 3D FreeSculpt. Elle permet de reproduire en quelques minutes tout type d'objet, à partir d'un simple transfert de fichiers via câble USB ou carte SD. Son écran LCD permet de suivre la progression de l'impression, tout comme sa large vitre transparente en façade qui permet d'assister à la création en direct de l'objet, couche par couche. L'imprimante est livrée avec un logiciel dédié pour Windows et OS X, une carte SD et un rouleau de plastique ABS (1kg). Ce modèle sera visible au salon 3DPrintShow (Paris, 15 et 16 novembre). Le constructeur propose, en option, des logiciels de capteurs 3D et d'édition. Ils sont utiles pour créer des modèles et les traiter. Le constructeur annonce déjà une version 2 pour 2014 ! C'est un marché qui évolue très vite ! site : pearl.fr



ATELIER DE GÉNIE LOGICIEL
PROFESSIONNEL

WINDEV®

DÉVELOPPEZ 10 FOIS PLUS VITE

10

NOUVELLE
VERSION

Java W PHP AJAX Windows Android iOS Applications natives

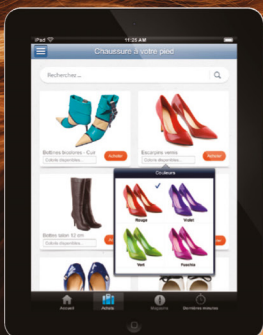
Venez découvrir

WINDEV 19 près de chez vous :
roadshow «TDF des versions
19» actuellement sur les routes.
Dates sur pcsoft.fr.

VERSION
EXPRESS
GRATUITE
Téléchargez-la !

Linux, Mac, Internet, Intranet,
Windows 8, 7, Vista, XP...,
Cloud, Android, iPhone, iPad :
vos applications sont compatibles

Environnement de développement professionnel, intégralement en français (logiciel, documentations, exemples). Développez facilement vos applications et vos sites. La facilité de développement avec WINDEV est devenue légendaire: vos équipes développent plus vite des applications puissantes, la qualité de vos logiciels est automatiquement élevée, le nombre de fonctionnalités automatiques est impressionnant. Vous délivrez plus vite vos logiciels, rapides et robustes.



DEMANDEZ VOTRE DOSSIER GRATUIT

260 pages - 100 témoignages - DVD Tél: 04.67.032.032 info@pcsoft.fr

www.pcsoft.fr



Fournisseur Officiel de la Préparation Olympique



Une vie à la (Silicon) Valley !

Chapitre 1 : Pourquoi y aller ?

Que vous soyez développeur, entrepreneur, innovateur ou disrupteur, ... tôt ou tard, vous risquez de côtoyer de près ou de loin ce haut lieu mythique des nouvelles technologies qu'est la Silicon Valley.

Trouvant ses origines dans les années 1970 avec le journaliste local Don Hoefler, la Silicon Valley doit son nom à la forte densité à cette époque, d'entreprises des domaines des semi-conducteurs et de l'informatique. La région était alors en effet plus connue pour ses vergers. Le terme Silicon vient de Silicium, matière déjà fortement exploitée dans les composants électroniques de l'époque.

Un passage par le « Computer History Museum » basé à Mountain View vous permettra de mieux comprendre l'histoire, tant de la région, que de notre domaine en général, qu'il soit du ressort du hardware ou du software. (Nous consacrerons un chapitre dédié au CHM dans un prochain chapitre tant le sujet est vaste et intéressant).



Aujourd'hui, la Silicon Valley est devenue un des hauts lieux mondiaux de l'informatique ou de l'information en son sens global, mais aussi de l'internet et des startups. Vous y retrouvez ainsi les sièges de sociétés mondialement connues comme Apple, basée à Cupertino, Google basée à Mountain View, ou encore Facebook basée à Menlo Park. L'université de Stanford y a également ses quartiers généraux à Palo Alto. Ainsi, la Valley est vue comme un endroit magique baigné d'inspiration, de pensée globale et d'action locale, de nouvelle vision du monde, de ruptures technologiques, de ruptures d'usages et aussi comme une source inépuisable d'innovations, de startups et de moyens financiers... Cette vision idyllique a forcément son revers que nous tenterons de vous faire découvrir parmi les facettes de la Valley.

> La Valley ? S'y rendre, pourquoi faire ?

Alors, « Pourquoi se rendre à la Valley ? » ... Les raisons peuvent être de tout ordre !



Tourisme, Famille, Inspiration, Travail, Startups, ... Quoi qu'il en soit, tout mouvement vers la Silicon Valley se doit d'être à minimum préparé tant vous avez de choses à voir, et d'opportunités à saisir lorsque vous y êtes.

Le mode touriste étant le plus simple, celui-ci ne requiert de vous qu'un passeport en règle et un billet d'avion que vous pouvez trouver à prix raisonnable. Une fois que vous y êtes, l'hébergement peut se faire à prix abordable au travers de formule type airbnb ou motels. Le must étant un contact local qui pourra vous accompagner et vous diriger vers les lieux importants à découvrir au sein de la Silicon Valley. Ne vous y méprenez pas, bien que proche, San Francisco n'est pas une partie intégrante de la Silicon Valley. Vous y trouverez cependant tout l'esprit et de nombreux incubateurs de startups... Une liste d'incubateurs de la région est reprise dans les liens attachés à cet article.

Hormis une démarche touristique, qui à elle seule en vaut grandement la peine pour tout passionné de nouvelles technologies, la région de la Valley reste l'endroit idéal pour venir chercher l'inspiration de vos futures entreprises (venture), innovations ou applications...

Effectivement, cette zone géographique est une forme condensée de ce qui se fait de mieux et de moins bien à l'échelle mondiale, un peu comme si vous aviez en un seul

endroit toutes les cultures, ethnies, classes sociales, leaders, suiveurs, innovateurs, idées, évolutions, ruptures, Toute la diversité du monde y cohabite afin de permettre à tout esprit novateur de pouvoir aller au bout de ses idées et inspirations... La Valley va vite, très vite et les décisions se prennent au rythme adapté ... en fonction de votre potentiel business et de sa rupture d'usages ou technologique.

> Les étoiles de la Valley ?

Contrairement à tout ce que nous pourrions croire, les stars du moment, celles et ceux qui sont sur les lèvres de tout un chacun, ne sont plus ou plus autant Google, Facebook ou autre startup du moment mais deux personnalités :

Marissa Mayer, Présidente et CEO de Yahoo!, ancienne employée Google avec le numéro 14 (non négligeable), qui est actuellement occupée à donner un coup de sang neuf à la marque Yahoo! et son marché. Elle a tendance à réussir tout ce qu'elle touche avec beaucoup de grâce et de délicatesse. Du relooking du logo de la marque au dernier renouvellement du service mail, en passant par des applications reconnues parmi les meilleures en termes



© World Economic Forum, swiss-image.ch/Photo Mafio Riess

DANS LA JUNGLE DU CLOUD, MIEUX VAUT CHOISIR LE BON PARTENAIRE.

b i z l o v Crédits photos : © Gettyimages / John Lund - John Lund / Blend Images / GraphicObsession

ET SI VOUS PASSIEZ À LA PUISSANCE CLOUD ?

VAR

Distinguez-vous de vos concurrents en valorisant votre offre.

SSII

Profitez d'infrastructures IaaS déployées en un clin d'œil.

ÉDITEURS

Passez au SaaS en vous appuyant sur notre savoir-faire.

PROGRAMME PARTENAIRE ARUBA CLOUD

- 2 niveaux de marque blanche disponibles.
- Gestion simple, souple et performante de l'infrastructure (publique, privée ou hybride).
- Modélisation et activation immédiate de votre datacenter virtuel dans le pays de votre choix.
- Interface client totalement personnalisable.
- Facturation en "Pay as you go".
- Finesse dans la gestion des droits utilisateurs.

Contactez-nous

Aruba, le bon partenaire pour bénéficier de la puissance d'un acteur majeur qui considère que chaque client, dans chaque pays, est unique. **MY COUNTRY. MY CLOUD.**


arubacloud.fr | TÉL : 0810 710 300
(COUT D'UN APPEL LOCAL)

d'expérience utilisateur, comme l'application Météo Yahoo!. Marissa est smart, a la gâchette facile, recrute à grande vitesse par rachat de startups innovantes ou d'équipes hors du commun, et peu de gens lui résistent ... Yahoo! a de beaux jours devant elle avec une Marissa Mayer à sa tête.

Elon Musk, CEO et CTO de SpaceX ainsi que CEO et Chief Product Architect de la non moins prestigieuse marque Tesla Motors. Cet entrepreneur visionnaire et audacieux, vu initialement comme un rêveur convaincu de pouvoir changer le monde avec des idées radicales, est clairement occupé à réussir l'une de ses missions avec la marque Tesla. Les voitures électriques sont de plus en plus nombreuses à la Valley, et il est impossible de faire une journée sans au moins croiser plus de 10 exemplaires de la marque de Fremont. Elon aurait-il trouvé la combinaison magique que de nombreuses autres marques et constructeurs de l'ancienne économie n'auraient pas su trouver jusque-là ? Force est de constater que les présentations d'Elon sont très convoitées et attendues par la presse internationale. Toutes les marques suivent le chemin tracé par Elon, les modèles électriques sont légion à la Valley, les stations électriques de recharge rapide poussent comme des champignons et toutes les entreprises majeures s'en équipent pour leurs employés.

> Les domaines clés de la Valley ?

Nous pourrions vous parler de domaines comme la mobilité qui est clairement le sujet central de toute startup de la valley, il va sans dire que le développeur Web ou d'application type PC n'est pas au goût du jour... La compétence mobilité est considérée comme fondamentale et ce pour toutes plateformes confondues avec une forte tendance iOS et Android...

Bien que les domaines restent nombreux au regard de la diversité des métiers et expériences utilisateurs, tentons de mettre le focus sur quelques domaines phares du moment : « le Big Data » et « la recherche par Ontologie » sont clairement deux domaines clés de la Valley et plus particulièrement de Palo Alto, épicerie de la région... Ces deux disciplines permettent d'apporter du « sens » à l'océan de données et d'accéder aux nouvelles générations de services ou moteurs de recherche. Les utilisateurs recherchent clairement de nouveaux usages et stratégies pour accéder à



l'information en surabondance et stopper l'effet d'inondation d'information !!!

L'Interaction Homme-Machine n'est pas en reste avec des tendances clairement affichées vers le motion sensing et l'après reconnaissance vocale afin de permettre, enfin, une réelle expérience utilisateur naturelle. L'écran sous forme simple et plus particulièrement la souris risquent de voir clairement leur champ d'exploitation de plus en plus restreint... Les limites apportées par ceux-ci ne permettant pas un usage complet à toute heure et en tout lieu.

Autant vous dire que la lutte dans ces domaines est rude et les startups nombreuses. Tout le monde veut tenter sa chance et « cracker l'usage ou le code » (expression locale). Les appelés sont nombreux, les élus beaucoup moins. C'est l'effet « Valley » : soyez prêt à tout miser et tout perdre car c'est une des qualités attendues ici, contrairement à d'autres régions... Même si vous êtes le meilleur de votre région d'origine, vous n'êtes potentiellement plus rien à la Valley. L'un des atouts majeurs de la Valley est clairement la capacité à mobiliser des moyens très importants pour atteindre une vision, une idée, une technologie, un produit ou la création d'une entreprise ... et ce quel que soit son stade d'évolution ! D'autres domaines mériteraient des articles entiers comme le « Bring Your Own Devices » (BYOD), le « Do It Yourself » (DIY), la révolution majeure de l'impression 3D, l'internet des objets connectés, le domaine médical sous toutes ses formes y compris par le séquençage génétique fortement supporté par l'un des fondateurs de Google, ... Et bien entendu, tous les sujets liés aux domaines de la Singularité Technologique : Génétique, Nanotechnologie et Robotique. Nos générations sont occupées à vivre une accélération technologique sans

précédent dans l'histoire de l'humanité, prenant de plus en plus une tournure de liberté dans les mains du consommateur. Ne reviendrions-nous pas enfin à un mouvement de liberté du consommateur plus que nécessaire et une explosion des anciens modèles qui ont plus que vécu jusqu'alors ? ... Pensons à Instagram versus Kodak ou encore Tesla versus Ford, ... Espérons-le ! Nous parcourrons également ce sujet lors d'un prochain chapitre. Des événements majeurs comme le TechCrunch Disrupt permettent ainsi de mettre en avant les nombreuses startups, projets et innovations de la Valley; ce type d'événement très fréquent ici est un fabuleux tremplin pour démontrer au plus grand nombre l'utilité de votre technologie ou produit et ainsi accélérer votre développement par le partenariat, la levée de fonds, la médiatisation, ... peut-être une excellente piste pour démarrer votre aventure et « une Vie à la Valley » !

> Et après ?

Après cette première ébauche d'une vie à la Valley, nous aborderons dans le prochain article les domaines disruptifs tels que la révolution automobile, la 3D ou encore les implications des travaux d'acteurs de Google autour du Global Brain tant convoité par l'industrie de l'information !

Greg - Geekman in SV
Oscaro.com & xBrainSoft Inside

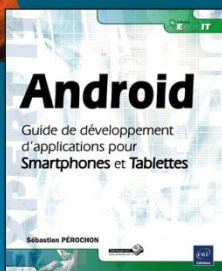
Annexes et liens :

- Computer History Museum : <http://www.computerhistory.org/>
- Stanford University : <http://www.stanford.edu/>
- TechCrunch Disrupt Event : <http://techcrunch.com/disrupt/>
- Incubateurs SV + US : <http://www.quora.com/Startup-Incubators-and-Seed-Programs/Which-incubators-exist-in-Boston-and-elsewhere-in-the-US>

OFFREZ-VOUS L'UNIVERS DU DÉVELOPPEMENT !

1 an 11 numéros
50€
seulement (*)

2 ans 22 numéros
79€
seulement (*)



+ Un «livre numérique»
au choix des Editions Eni

Version numérique : 29,26 €

Version numérique : 40,5 €

Version numérique : 29,26 €

Valeur des 11 numéros du magazine : 65,50€ + Valeur du livre : 29,26 € à 40,50 € = Valeur totale : 94,76 € à 106 €

réservée à la France
Métropolitaine

(*) Tarifs France métropolitaine

Toutes nos offres sur www.programmez.com

Oui, je m'abonne

à retourner avec votre règlement à
Programmez, 17 route des Boulangers 78926 Yvelines cedex 9

Attention : l'offre avec livre Eni est limitée et réservée à la France métropolitaine

- ☐ **Abonnement 1 an au magazine + 1 « Livre Numérique ENI » :** 50 € (au lieu de 65,45 €, prix au numéro + valeur du « livre numérique ») ☐ Abonnement seul : 49 €
☐ **Abonnement 2 ans au magazine + 1 « Livre Numérique ENI » :** 79 € (au lieu de 130,9 €, prix au numéro + valeur du « livre numérique ») ☐ Abonnement 2 ans seul : 78 €

Livre à choisir : ☐ Html 5 CSS 3 et JavaScript ☐ Objective-C ☐ Android

Détails sur www.programmez.com/abonnement.php

Tarifs France métropolitaine

☐ M. ☐ Mme ☐ Mlle Entreprise : _____ Fonction : _____

Prénom : _____ Nom : _____

Adresse : _____

Code postal : _____ Ville : _____

Tél : _____

E-mail : _____ @ _____

☐ Je joins mon règlement par chèque à l'ordre de Programmez !

☐ Je souhaite régler à réception de facture

(Attention, e-mail indispensable pour les archives sur internet et offre Eni)

Stéphane est tombé dans la **portion magique** du développement

Stéphane Graziono est tombé sur sa première ligne de code à 15 ans. 14 ans plus tard, il est toujours passionné et aime ce métier, parfois exigeant. Comme nous, il a sérieusement débuté son apprentissage avec l'excellent Atari STE !



Comment es-tu tombé dans l'informatique et plus spécialement dans le développement ?

J'avais 7 ans, mon père avait acheté un ATARI STE, et quelques années plus tard un PC 386. Il avait bien vu que je m'y intéressais, du coup il me demandait de faire les disquettes de boot (autoexec.bat/config.sys) pour adapter la mémoire pour certains jeux. Comme j'étais aussi très branché jeux-vidéo (j'ai eu la première ATARI avec Space Invaders !) je me suis rapidement demandé comment on pouvait faire des jeux. Je flânais dans les rayons des bouquins d'Informatique, je ne comprenais absolument rien de ce que je pouvais y lire, ne sachant pas vraiment quoi chercher, mais je me souviens d'un gros bouquin Borland C++ que j'avais commencé à lire et qui m'avait intrigué ; ceci-dit j'avais compris qu'il y avait des logiciels chers à payer donc j'avais mis l'idée de côté. J'ai finalement commencé à coder au lycée en 2nde, avec ma première calculatrice programmable Texas-TI80, la doc était plutôt bien faite et le Basic était très accessible. J'ai rapidement pu faire du Quick-Basic sous Dos, puis du Visual Basic grâce à un cours d'info au lycée. Un pote du lycée (Mathieu Lamarche, qui est aujourd'hui mon meilleur ami) était fou de technologies et m'a montré son programme en C avec les API Win32 pour créer et gérer une fenêtre, me vantant les performances du code C, au

“ Développer ? un challenge quotidien ! ”

détriment de la facilité. J'ai lâché directement le VB pour le C++ et on s'est mis à faire de l'OpenGL et DirectX, puis notre premier petit jeu de plate-forme « Poppy's Adventures », dont l'héroïne n'est autre qu'une petite souris rouquine que j'avais au lycée. Quelques années plus tard, j'ai voulu faire du Web, je trouvais ça vraiment cool de pouvoir montrer son savoir-faire au monde entier, et j'ai sorti mon premier site web sur la basse (mon autre passion), c'était il y a 10 ans, et il est encore ouvert aujourd'hui (www.slappyto.net). Après des débuts d'études difficiles en électronique et informatique industrielle, j'ai fait mes études d'Informatique en alternance dans une société spécialisée en RFID - TagProduct sur Lyon - où j'ai programmé pas mal de terminaux portables (Windows CE), les ancêtres des tablettes. Aujourd'hui c'est le monde des smartphones/tablettes qui m'intéresse, et je sors mon premier jeu en tant qu'indépendant en Novembre « Poppy Kart » sur Android/iOS/ Windows Phone 8/Windows 8 qui est pour moi un grand retour aux sources avec OpenGL et DirectX !

Pour toi, qu'est-ce qui fait que l'on aime toujours et encore le développement, la technique ?

Pour moi c'est un challenge quotidien, on a une solution à trouver, une boîte à outils

géante, et des contraintes (pour ma part, je garde toujours la même directive : mon code doit être rapide ET le plus simple possible). Les nouveaux appareils et OS rendent le métier encore plus passionnant, bien que ça le complique aussi. Il y a aussi l'effet communauté : on a l'impression d'appartenir à un groupe, on essaye de s'entraider. Mais le challenge reste vraiment mon leitmotiv, et j'ai même des moments d'euphorie quand j'arrive à doubler la rapidité de mon code. J'ai baigné dans cette optique depuis mes débuts ; J'ai vraiment eu beaucoup de mal avec les langages managés à leur sortie ; Java était d'une lenteur infernale sur des petites configurations.

Plus tard j'ai pris le pli quand les machines étaient devenues plus puissantes. J'aime aussi l'idée d'avoir un code qui tourne sur toutes les plates-formes, moyennant quelques sucres. Je suis vraiment un technicien pur et dur : écrire une application en 9 couches dans les langages managés me fait froid dans le dos – sauf si on me donne la main pour optimiser la base de données et les requêtes !

Tu as gardé un regard très geek : gadget, veille techno. c'est important pour ton job et ta passion ?

Je pense être né à la bonne époque, j'ai quasiment tout vu passer que ce soit ordinateurs ou consoles de jeux ; c'est important pour moi d'avoir ce bagage, ça me donne

une direction mais du coup j'ai un avis bien tranchant sur certaines technos. Je consulte tous les jours les news Geek pour savoir comment le monde techno avance, ça permet aussi d'avoir une vision globale, même si, comme les infos, c'est à prendre avec des pincettes. Je trouve que l'engouement des gens pour les nouvelles technologies est rassurant pour notre futur, c'est vraiment un métier d'avenir, même si les écoles d'Informatique ne sont pas spécialement remplies.

Etre développeur n'est pas toujours facile : pression, évolution constante, frustration des projets et des "chefs", c'est quoi pour toi être développeur aujourd'hui ? Le job a-t-il changé depuis tes débuts ?

Le métier de développeur est un métier très difficile mentalement. Il a beaucoup changé car beaucoup de nouveaux acteurs sont venus se greffer. Il faut une équipe de n personnes pour faire le moindre projet : écrire une proposition commerciale, écrire des spécifications ; il faut des designers, des intégrateurs, des ergonomes, des codeurs, des testeurs, etc. encadrés par des chefs de projets.

On se retrouve confronté à un monde où la communication est souvent bien complexe et on passe plus de temps à répondre à des mails et faire des réunions qu'à faire de la réalisation. Les moins techniciens d'entre nous apprécieront le côté social. Pour ma part, le job a changé puisque je suis passé indépendant, il y a 2 ans - avec mon frère jumeau Franck Graziano - pour avoir la liberté de me donner à fond dans ma passion, avec d'autres contraintes - la gestion d'une société - et honnêtement, je me sens bien plus à ma place !

Et en dehors du boulot, qu'est-ce que tu aimes faire ? Comment trouves-tu l'équilibre entre travail, vie privée, passion, famille ?

Je suis bassiste dans un groupe de rock alternatif « Bird is the word » sur Lyon, ça permet de se retrouver avec les potes et on passe vraiment de bons moments sur scène. C'est un paradoxe avec le métier de développeur puisqu'il faut se montrer ! Je ne pense pas avoir trouvé encore un équilibre très sain, je devrais passer moins de temps à coder, mais n'ayant pas encore une vie familiale très chargée, j'en profite un peu - j'en profite pour faire un bisou à ma copine qui supporte ça ! Ceci dit j'essaie de sortir le plus possible, même une soirée geek ce n'est pas du boulot vu qu'on est avec ses potes !

Peux-tu nous présenter ton quotidien en quelques mots ?

Je commence à 9h, je vérifie mes mails, je regarde ma todo-list partagée sur Google Drive (en général elle est gigantesque, il y a tout de marqué jusqu'à la fin du projet !) je code jusqu'à 13h, je reprends à 14h et je finis à 20h. Si je suis en période intense (motivé) je reprends le travail de 21h jusqu'à tard, disons 1h - Je travaille vraiment mieux la nuit, ou alors très tôt le matin, ma concentration est très sensible au calme sonore. On fait souvent des points avec mon frère, savoir où en est l'autre (il fait la partie multimédia) pour avancer le plus possible synchronisés.

Bon ça fait des journées de 10h à 14h, c'est plutôt chargé mais vu que j'aime ce que je fais en ce moment ça passe bien plus vite que mes anciennes 37,5h.

Comment vois-tu ton job évoluer ?

Mon job évoluera en synchronisation avec l'évolution de ma société, donc si tout va bien je devrais déléguer un peu le code dans l'avenir, au profit de l'architecture, mais je garderai toujours un œil et une main sur le code, trop passionné pour fermer les yeux ! J'aimerais bien aussi si l'occasion se présente enseigner en école en tant que vacataire, je me souviens avoir eu des profs qui nous avaient appris des trucs formidables, et je trouve ça vraiment cool.

Découvrez quelques applis de Stéphane : site : <http://apps.webbrox.fr>

la page FB de PoppyKart

www.facebook.com/PoppyKart

Les conseils de Stéphane

Pour les étudiants :

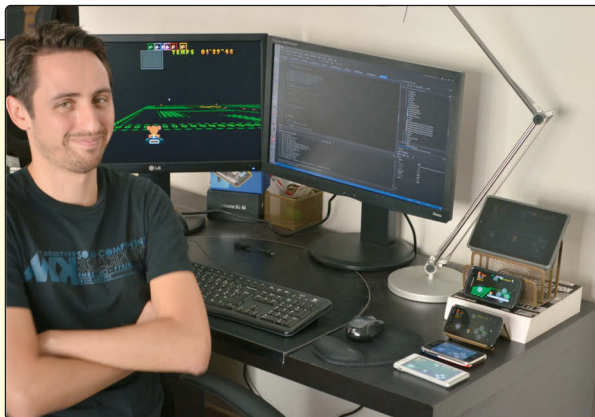
- faites du C et C++ pendant plusieurs années, ça vous sauvera toute votre vie et ça vous donnera d'excellents réflexes : on programme des machines dont quasiment tout est basé sur la mémoire et l'adressage, ne pas savoir comment fonctionne un pointeur ou la mémoire, c'est un peu honteux et pire ça donne de très mauvais réflexes de programmation dans les langages évolués/managés.
- Nommez vos variables convenablement, préfixez les avec leur type, ça facilite vraiment la relecture.
- Ne pas sous-estimer le métier de programmeur, c'est un métier très intéressant et surtout très créatif, malgré la complexité.
- Et un autre très important basique : lire et comprendre l'anglais ! Vous ne trouverez jamais de documentation en français et si vous en trouvez une : vous ne la comprendrez pas !

Pour les développeurs :

- Apprendre à faire des requêtes avancées sur Google (guillemets, moins etc) et fouiner sur le web vous fera gagner beaucoup de temps, et bien évidemment partager sur votre blog vos tips s'ils sont introuvables !
- Ecrivez du code que les autres pourront comprendre. Privilégier les choses simples qui marchent. S'il y a un très bon codeur dans une équipe de développeurs moyens, son départ pourrait compromettre la finalisation d'un projet !
- Communiquer rapidement avec vos chefs quand vous sentez une grosse contrainte technique arriver
- Et sinon étant un gourou des expressions régulières, je ne peux que vous conseiller de les apprendre, vous économiserez 90% de votre temps à jouer avec les chaînes de caractères, et la gymnastique mentale est très agréable !

Mon bureau

«J'ai un PC portable pas ultra puissant (i3 2.4Ghz) avec 8Go de RAM, mais je lui ai mis un disque dur SSD - le maillon faible de l'ordi - du coup, on ne ressent aucune gêne ! Je tourne sous Windows 8 Pro avec VS Express 2012, Eclipse, et SVN. J'ai aussi un MacBook Pro pour coder iOS. Et pour la partie mobile j'ai un peu de tout : Nokia Lumia 620, Sony Xperia-SP, Samsung Galaxy S, Nexus 7, iPhone 3GS. C'est important pour moi d'avoir aussi des appareils « bas de gamme » pour tester les performances ; après, on peut lancer une application sur smartphone tranquillement sur le marché. Vue ma passion pour la musique j'en écoute beaucoup (au casque pour ne pas embêter mon collègue) mais quand les algorithmes sont compliqués, je la coupe. Pour l'instant je travaille de chez moi ce qui a autant d'avantages que d'inconvénients.»



TF Server contre TF Service : le combat !



> Team Foundation Server, 1er du nom ! [Fig.1]

Pour rappel, Team Foundation Server est un outil inclus dans le package de Visual Studio permettant de gérer :

- le code source,
- les builds,
- les performances,
- le suivi du projet,
- les tests et environnements de tests,
- Etc.

Lors des TechDays 2013, de nombreux intervenants ont confirmé qu'avec la nouvelle version 2012 de TFS, Microsoft cherche à réunir tous les intervenants autour de la réalisation d'un projet par le biais de TFS. L'objectif de cet article n'est pas de vous prouver l'intérêt de la nouvelle interface disponible depuis la version 2012 de Visual Studio. Si cela vous intéresse, je vous conseille de la découvrir à travers la présentation des TechDays de Février 2013 : <http://goo.gl/aPBvVa> Ce bel outil permet de gérer efficacement la réalisation de son nouveau produit. Que vous pratiquiez, comme moi, le Scrum ou le Kan-Ban ou que vous préférerez travailler dans un cadre CMMI, il vous suffit de choisir un template adéquat afin d'avoir des artefacts adaptés à vos besoins.

> Le nouveau TFS : Team Foundation Service (TF Service) [Fig.2]

En 2012, Microsoft a lancé son nouveau produit : Team Foundation Service souvent appelé TFS, ce qui n'est pas sans créer la confusion avec la version serveur également appelée TFS. Pour faire simple, TF Service est la version Cloud de son grand frère TF Server. Microsoft fait régulièrement évoluer TF Service. De plus en plus, toutes les fonctionnalités présentes dans la version Server apparaissent dans le Cloud. Inversement, la plupart des fonctionnalités que l'on trouve exclusivement dans le Cloud se retrouveront

Microsoft propose depuis 2012 une nouvelle version de TFS : Team Foundation Service (TF Service). Quelle version, TF Service ou TF Server, est la plus intéressante ? Avec l'arrivée de Visual Studio 2013 et l'amélioration continue de Team Foundation Service, il est pertinent d'examiner ce duel fratricide.

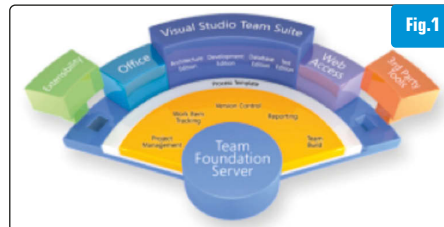


Fig.1

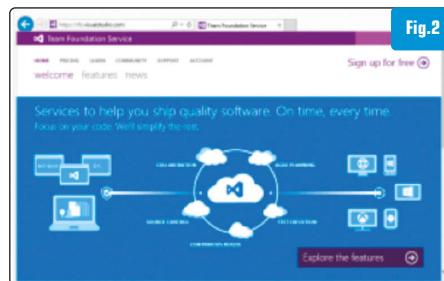


Fig.2

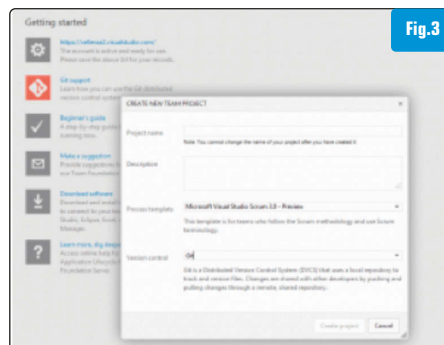


Fig.3

dans la version Server. Nous avons depuis le début une pré-version de Team Foundation Service : la gestion des codes sources et l'interface web pour gérer son projet. Le tout compatible nativement avec Visual Studio 2012 et avec la version 2010 à l'aide d'un Service Pack. TFS continue son évolution. Vous pouvez maintenant gérer (en version bêta) vos builds, vos tests sur cette version et vous pouvez utiliser Git comme contrôleur de code source [Fig.3].

> TFS versus TFS

Finalement, nous avons une version Cloud qui continue à s'améliorer et une version Server déjà très solide. Autant le dire tout de suite, dans ce duel qui oppose la version Server à la version Cloud de TFS, il n'y a pas

de vainqueur ! Les deux se valent et votre choix se fera en fonction de votre environnement et de vos besoins.

> Team Foundation Server

La version Server a pour moi deux avantages indéniables :

1. Vous n'êtes pas satisfait du Template que vous utilisez ? Le Workflow ne vous convient pas ? Vous voulez ajouter des champs ? Aucun problème, il suffit d'installer Team Foundation Power Tools et Pimp your TFS !
2. Vous pouvez utiliser les comptes utilisateur de la société contrairement à TF Service qui se base sur live id.

> Team Foundation Service

La version Service a, elle aussi, ses avantages :

1. Accessible facilement, rapidement et partout. C'est la promesse du Cloud et TFS la respecte : où que vous soyez, vous pouvez partager votre code ou suivre l'avancement du projet. Quelques clics suffisent pour créer votre compte et inviter vos collaborateurs dans votre espace TFS.
2. Gratuit jusqu'à 5 utilisateurs : parfait pour tester et se forger un avis. Pour cela il vous suffit d'utiliser votre adresse Live que vous utilisez déjà sur tous les sites de Microsoft.
3. Il évolue très rapidement. Chez Microsoft, ils nous prouvent une fois encore qu'ils savent ce qu'est l'agilité et les itérations. Les livraisons sur TF Service sont plus nombreuses car les sprints sont plus petits, mais TFSer a aussi évolué puisqu'il y a eu 3 mises à jour depuis la dernière RTM.

QUE CHOISIR ?

Suivant la taille de votre projet et son legacy le choix se fera naturellement. Vous travaillez sur un gros projet et vous avez besoin de pouvoir adapter votre outil à vos besoins ? La version TF Server vous permettra de faire cela. Vous pourrez même maîtriser et amé-

liorer vos builds et tout ce dont vous avez besoin pour assurer la qualité de votre produit et l'intégration en entreprise sera facilitée par cet outil. Si vous êtes plutôt sur des petits projets, foncez vers la version Service. Pas de serveur à gérer et un système de gestion de projet et de code source de qualité à moindres frais s'offre à vous !

Si je devais résumer je dirais que :

- TF Service est très bien pour du communautaire : petits projets, expérimentation, coding dojo, etc.
- TFSer est fait pour s'adapter aux contraintes et exigences du développement pour des grosses structures et de grosses entreprises.

Enfin, sachez que quel que soit votre choix, vous pouvez facilement gérer votre backlog et tous vos Work Items avec Excel. Pour cela il vous suffira d'installer l'add-in de Team Foundation dans Excel afin de pouvoir synchroniser vos feuilles Excel avec votre TFS.

QUOI DE NEUF DANS TF SERVER 2013 ?

Comme je vous le signalais dans l'introduction, une nouvelle version de TFS est en cours de test depuis la présentation de la Build 2013 de Microsoft en Juin 2013. Mais quelles sont donc les différences notables ?

> La gestion des Portfolios Agiles

Visuellement, on peut se rendre compte que le portail web se voit agrémenté de code couleur pour rendre l'arborescence plus lisible [Fig.4]. Ensuite, l'ajout des nouveaux work items (les Initiatives et les Goals) permettra au product owner et au Top Management d'avoir une meilleure vision de l'évolution et l'affinage de backlog tout au long du projet. Enfin, afin de proposer une gestion du type Scrum de Scrum, TFS recycle son champ Area. Rappelons qu'on pouvait le manipuler depuis toujours dans les work items afin de l'utiliser comme filtre dans le portail web (Team Web Access) et obtenir ainsi plusieurs sous-backlogs.

Ces évolutions ne payent pas de mine à première vue mais sont toutefois de très bonnes initiatives de Microsoft pour fédérer toujours plus les participants d'un projet autour de TFS. Le product owner et toutes les parties prenantes pourront facilement visualiser l'évolu-

tion du projet et la répartition des tâches par équipes sur le portail web de TFS.

> Le Contrôle de Version

Nouvel ajout déjà présent dans TF Service : l'intégration native de Git. Ceci offre ainsi une alternative communautaire au serveur industriel de TFS. Dans la page d'accueil du Team Explorer, vous pourrez facilement gérer les Pending Changes, les Commits, etc. De plus, il sera possible de commenter le code directement depuis Team Web Access [Fig.5].

> Rafraîchissement du Team Explorer

Petite amélioration graphique dans la version 2013 du Team Explorer qui devient beaucoup moins austère que la version 2012. Pour cela, il suffisait d'ajouter des icônes et tout de suite cela devient plus lisible [Fig.6].

> Import / Export

Microsoft offre la possibilité d'importer et d'exporter les données de TF Service vers TF Server. Je trouve cette option très intéressante ! Cela vous permet de commencer à développer et gérer votre projet sur la version Service pour ensuite migrer vers la version serveur quand votre projet commence à prendre de l'ampleur. Super initiative pour

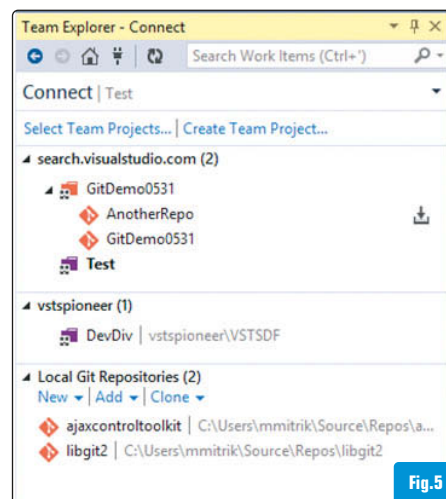


Fig.5

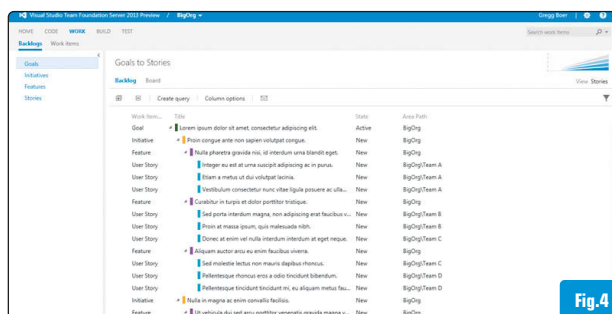


Fig.4

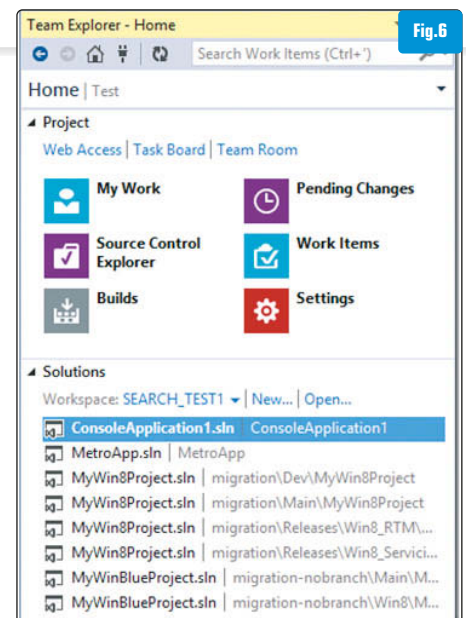


Fig.6

expérimenter un nouveau projet test avec un nouveau template TFS ou avec un nouveau langage avec quelques collaborateurs...

> Les autres évolutions à suivre

Tests : Possibilité d'éditer des scénarios de test directement depuis Team Web Access et Visual Studio. De quoi remplacer Test Manager dans certains cas ?

Release Management : La solution InCycle InRelease a été acquise par Microsoft et sera probablement bientôt totalement intégrée à TFS. Elle permet de gérer finement les plans de release, via l'organisation des builds, changesets et work items. A tester et surtout à surveiller pour voir son évolution dans le temps.

Collaboration : Chaque équipe TFS se voit attribuer un espace collaboratif. C'est la Team Room. Cet espace a pour but de centraliser et archiver les interactions entre les membres de l'équipe : changesets, builds, revues de code, dialogues, etc. L'objectif est de faire de cet espace la page d'accueil des membres de l'équipe.

Alors cette version 2013 ?

Au final, la version 2013 ne simplifie pas le choix entre les 2 versions (Service ou Server). Microsoft confirme que toute amélioration apparaissant dans la version Cloud finit par arriver dans la version Serveur et vice versa. Il est néanmoins intéressant de noter que Microsoft continue à améliorer ses solutions afin de réunir tous les cœurs de métiers à tous les niveaux autour de ses outils. Accessibilité et complémentarité, cela me donne même parfois envie de brancher du java sur un TFS...

Arnaud Villenave - XEBIA IT ARCHITECTS

Les nouveautés de Visual Studio 2013

En juin dernier s'était tenue la **Microsoft Build Developer Conference**, durant laquelle les premières nouveautés de Visual Studio 2013 furent dévoilées. Il est maintenant temps de faire un dernier tour de ce qui nous attend. Visual Studio 2013 sortira officiellement le 13 novembre prochain.

Au programme, une sélection des nouvelles améliorations de l'IDE, suivi d'un passage en revue des nouveautés côté langages et technologies, et pour finir, un aperçu des innovations concernant l'ALM !

AMÉLIORATION IDE

> Productivité

Avec cette nouvelle version de Visual Studio, Microsoft a fait preuve d'écoute envers ses utilisateurs en y implémentant quelques-unes des plus populaires **User Voices*** de chaque domaine. [*Le site des User Voices regroupe des idées d'améliorations proposées par les utilisateurs et pour lesquelles il est possible de voter <http://visualstudio.uservoice.com>]

Ainsi, l'une des premières améliorations constatée est visible directement au lancement de Visual Studio : la page de démarrage a été simplifiée, le design a été revu et c'est maintenant un **thème bleu** avec des icônes bien plus colorées qui nous accueille dès l'ouverture.

La seconde grosse nouveauté se matérialise par une fenêtre qui s'affiche à votre premier lancement de Visual Studio. Celle-ci vous permet de vous connecter à votre compte Live, afin de **synchroniser automatiquement votre configuration personnalisée dans le Cloud**. Si, par le futur, vous aviez à utiliser Visual Studio depuis un autre poste de travail, vous n'auriez alors qu'à ressaisir vos identifiants Live pour récupérer instantanément votre configuration.

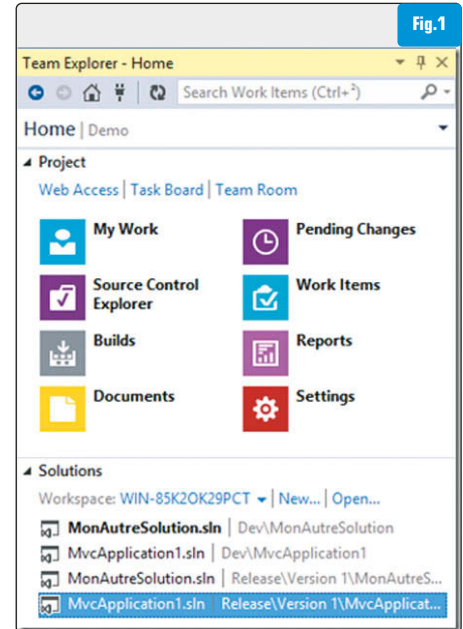
Le **Team Explorer**, soit la fenêtre de connexion à Team Foundation Server (dont les nouveautés sont détaillées plus tard dans cet article), fait également peau neuve : une grille d'icônes colorées permet dorénavant d'accéder aux classiques sections de TFS (Code source, builds, éléments de travail, etc.). On retrouve également un comportement similaire à l'ancien Visual Studio 2010 avec la possibilité de détacher du Team Explorer les fenêtres des *Modifications en attente* et de l'*Explorateur de Build*. Il est à noter également l'apparition d'une nouvelle

section bien pratique appelée *Solutions*, qui liste automatiquement les solutions disponibles dans votre *espace de travail* [Fig.1].

Comme à l'accoutumée, une bonne partie des fonctionnalités apportées par les **Visual Studio Productivity Tools** sont intégrées dans l'installation standard. On y retrouve la fermeture automatique des accolades, la possibilité de déplacer des lignes de codes avec des raccourcis clavier, mais également la *barre de défilement* avancée. Cette dernière apporte le fameux mode *map* qui remplace avantageusement la barre de défilement par une miniature de l'intégralité de votre fichier, et à partir de laquelle vous pouvez afficher un aperçu du code source.

Et pour finir, deux dernières innovations qui se sont révélées particulièrement intéressantes : Le **Peek Definition** et le **Code Lense** : Le **Peek Definition** s'apparente au déjà très connu *Go to definition*, à la différence que ce premier permet de naviguer vers une définition tout en restant dans le contexte initial qui l'a appelée. En effet, au lieu d'utiliser le simple raccourci F12, il faut presser ALT+F12 pour faire alors apparaître ses résultats dans une petite fenêtre appelée *Peek Window*. Si on continue d'utiliser le *Peek Definition* à partir de cette même fenêtre, de petits cercles de navigation apparaissent et permettent ainsi de savoir à tout moment combien de niveaux de définition ont été parcourus, mais également de revenir très facilement à un niveau précédent.

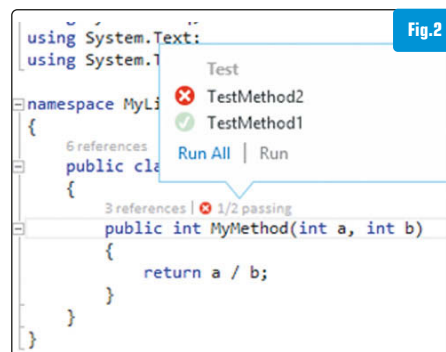
Le **Code Lense** quant à lui prend la forme d'annotations affichées directement dans l'éditeur de code, au-dessus de chaque



méthode. On y trouve le nombre de références faites à cette méthode (et la possibilité de les lister et d'y accéder), un statut global des tests unitaires qui testent cette méthode (avec la possibilité de visualiser les tests en échec, et de les relancer directement depuis le *Code Lense*), la dernière personne ayant modifié le corps de cette méthode, ainsi que le nombre de modifications qui y ont été apportées (avec l'historique TFS de cette méthode, des options accessibles en quelques clics pour comparer les différentes versions, etc.) [Fig.2 et 3].

> Performances

De nombreuses optimisations des performances sont également présentes dans cette nouvelle version. Il suffit simplement de passer du Framework 4.5 au Framework



4.5.1 pour bénéficier automatiquement d'améliorations apportées à l'exécution du Framework.

On apprécie également le changement apporté au **chargement des projets** d'une solution. Celui-ci s'effectue dorénavant **de manière asynchrone**, en restaurant en premier lieu votre espace de travail avec vos différents fichiers ouverts. Cela permet de commencer à travailler sur ces fichiers dès l'ouverture de la solution, tout en laissant à Visual Studio le soin de charger en tâche de fond les autres projets qui la composent.

> Debugging

La partie débogage dans Visual Studio n'est pas en reste. À commencer par l'ajout du **débogage asynchrone** pour les applications **Windows Store 8.1**, ainsi que l'implémentation de l'une des fonctionnalités les plus réclamées des User Voice : le **x64 Edit and Continue**. Celui-ci vous permet, en pleine session de débogage, de modifier votre code source, puis de poursuivre l'exécution du débogage sans devoir relancer l'application, ce qui permet un grand gain en productivité. Enfin petite nouveauté qui ravira beaucoup d'entre nous, la **valeur retournée par une méthode** est maintenant directement accessible dans la fenêtre *Autos* de débogueur.

AMÉLIORATIONS LANGAGES ET TECHNOLOGIES

> Windows 8.1 (Store Apps, Blend)

La sortie d'une nouvelle version de Visual Studio correspondant généralement à la sortie d'une nouvelle version de Windows, l'une des principales nouveautés, si ce n'est la principale nouveauté de ce Visual Studio 2013, est bien entendu de fournir la possibilité de **développer des applications Windows Store 8.1**.

Pour commencer, on distingue de nombreuses **améliorations au niveau des éditeurs**, que ce soit **HTML/JS ou XAML**. On constate par exemple une bien meilleure IntelliSense du côté de l'éditeur XAML de Visual Studio, avec le support de l'auto-complétion sur les *Data Bindings* et les *Resources*. L'outil de design Blend n'est pas non plus oublié. Celui-ci offre un bien meilleur support du HTML/JS avec, entre autres, la capacité à créer des animations CSS depuis la *timeline editor*, de configurer le *Data Binding*, de rajouter des *Behaviors* en JavaScript, etc.

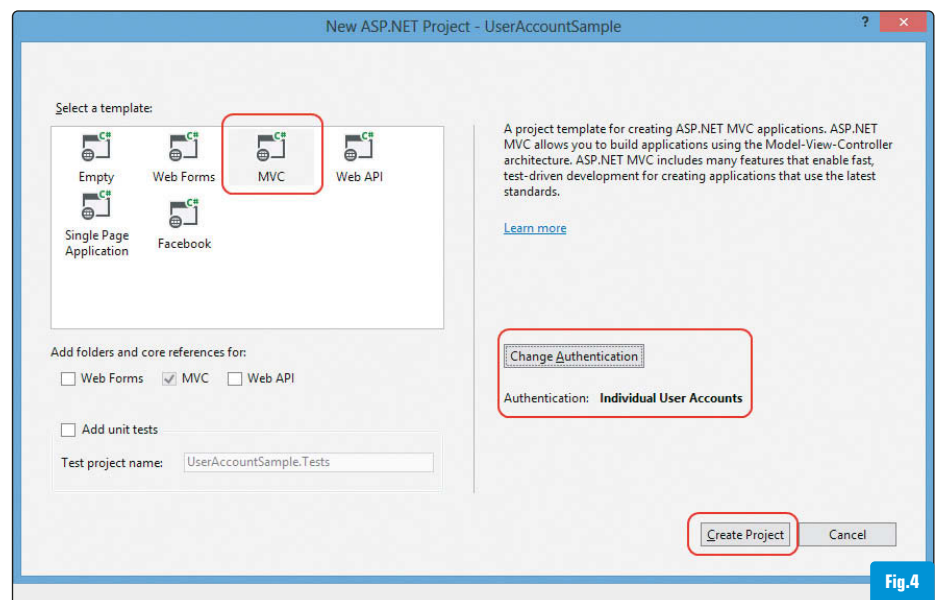


Fig.4

Windows 8.1 apporte également son lot de **nouveautés concernant les contrôles graphiques** disponibles. On y trouve non seulement des mises à jour de contrôles existants avec, par exemple, la correction du rendu de la *WebView* XAML, mais également l'apparition de nouveaux contrôles HTML/XAML tels que le contrôle *Hub*, ainsi que l'*AppBar* Command/CommandBar.

Côté Windows Store, un **nouveau type de packaging** permet de séparer plus facilement les ressources principales des ressources additionnelles. On peut ainsi fournir des packages de ressources additionnelles suivant la localisation (*du texte en Français, des images contenant du texte en Français, etc.*) et ainsi permettre d'obtenir une **App plus légère** à télécharger. Il faut également noter l'apparition d'un **nouveau design** pour le Windows Store qui permet de mettre plus facilement en avant les autres Apps publiées par le même auteur. Enfin, une petite nouveauté qui va bien faciliter la vie des développeurs d'Apps, la **mise à jour automatique des Apps** est activée par défaut sur Windows 8.1.

Pour conclure sur ces nouveautés Windows 8.1, une bonne nouvelle pour les équipes qui ont besoin d'un cycle de vie plus automatisé pour leurs applications :

Les **Coded UI Tests** (*tests d'interface automatisés*) **supportent maintenant les tests automatiques sur les applications Windows Store**. Il est ainsi possible de lancer votre application, d'identifier ses différents contrôles, ainsi que de simuler des Gestures afin de reproduire une manipulation d'interface tactile.

> ASP.net 4.5.1, JavaScript, Web

C'est durant la session *What's new in ASP.NET and Visual Studio 2013* tenue lors de la Microsoft Build 2013 que Scott Hanselman a pu dévoiler les nombreuses nouveautés apportées à la partie Web.

En premier lieu, on remarque la présence d'un **unique template de projet web**, appelé sobrement ASP.NET Web Application. Ce n'est qu'après avoir sélectionné celui-ci que s'ouvre un assistant, permettant de configurer la structure de démarrage que possèdera votre site web. Il est ainsi très facile de composer un projet reposant sur la technologie Web Forms, MVC ou Web API, voire même les 3 technologies en même temps [Fig.4].

Une autre fonctionnalité maintenant étendue à tout type de projet web : **L'ASP.NET Scaffolding**. Celui-ci permet de **générer automatiquement un ensemble de fichiers permettant d'interagir avec un modèle de données**. On peut par exemple obtenir en quelques clics un contrôleur accompagné de différentes vues (CRUD) avec toutes les dépendances nécessaires, et ce, même si la configuration de projet initialement choisie était de type Web Forms.

L'une des nouveautés majeures, pour les développeurs web, apportée par cette nouvelle version de Visual Studio reste sans doute la nouvelle fonctionnalité du **Browser Link**. Ce dernier permet de connecter un ou plusieurs navigateurs web à votre IDE. Après modification de code source, il suffit de cliquer sur le bouton de rafraîchissement pour **mettre à jour instantanément tous les**

navigateurs connectés. Cette fonctionnalité repose sur le tout récent ASP.NET SignalR qui permet d'ajouter des fonctionnalités de temps réel à des applications web.

Pour finir avec ces nouveautés web, il faut noter l'arrivée d'**ASP.NET Identity**, qui se positionne en tant que nouveau système d'authentification. Il propose une meilleure intégration avec les plus récents fournisseurs d'identités, que ce soient des identités récupérées via des réseaux sociaux (*comptes Windows Live, Facebook, Google, Twitter*), ou encore via le tout nouveau Windows Azure Active Directory.

> VC++

Les développeurs C++ n'ont pas été laissés à l'abandon avec cette nouvelle version de Visual Studio. À l'inverse, l'intention récente de Microsoft de ramener l'outillage C++ au niveau de l'outillage des autres langages est très perceptible.

Avec Visual Studio 2013, l'éditeur C++ possède de meilleures options de formatage pour la syntaxe du code. Auto-complétion des accolades/guillemets/parenthèses fermantes, ainsi que d'autres éléments tels que les commentaires multi-lignes.

Le **débogage asynchrone** des applications Windows Store ainsi que le *Just My Code* sont également disponibles.

On note également un gros effort fourni pour coller aux standards de l'**ISO C++ 11**. La plupart de ses fonctionnalités y sont maintenant supportées.

> SharePoint, Office 365, Azure

De nombreuses améliorations sont également présentes sur d'autres produits. Parmi celles-ci on peut citer l'intégration des Azure Mobile Services et Azure Websites, manipulables directement depuis Visual Studio. Il ne faut pas non plus oublier l'ajout du support de l'ASP.NET MVC en tant qu'Apps développée pour SharePoint 2013

ALM

> Team Foundation

Avec cette nouvelle version de Visual Studio s'associe tout naturellement la nouvelle version 2013 de Team Foundation Server. Cette dernière ne se contente pas de regrouper toutes les mises à jour qui sont déjà accessibles via Team Foundation Services, sa version hébergée dans Windows Azure, mais apporte également son lot de nouveautés.

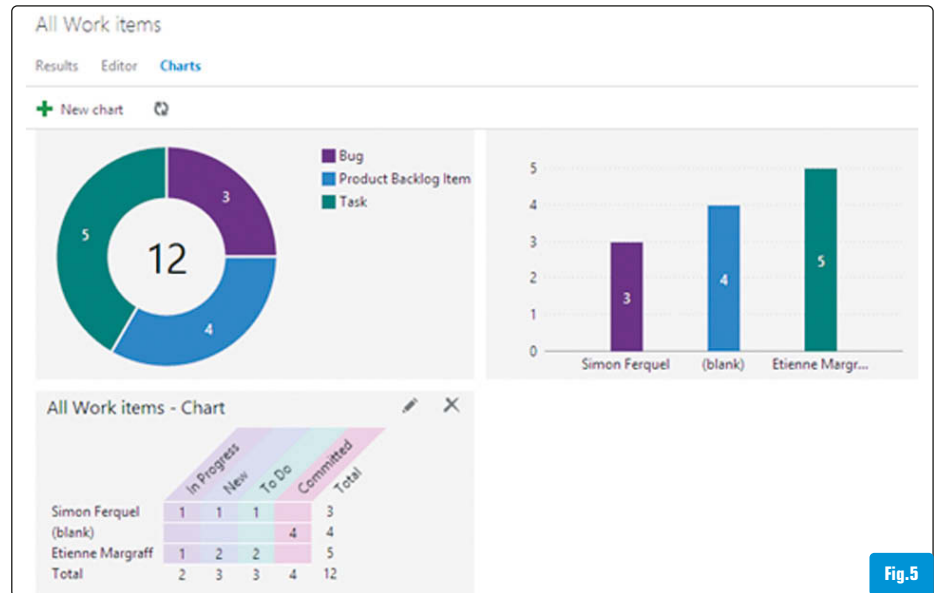


Fig.5

Attendu par de nombreux développeurs, le **support de GIT avec Team Foundation Server** est maintenant disponible **on premise** dans sa version 2013. Depuis Visual Studio, il est possible de se connecter aussi bien au respository GIT d'un projet d'équipe TFS, mais également à tout repository GIT standard tel que ceux présents sur GitHub. Les différentes actions permettant d'interagir avec le repository sont accessibles directement via l'interface graphique du Team Explorer.

Autre nouveauté, cette fois-ci concernant la gestion des éléments de travail : la possibilité de définir des **portfolio backlogs**. Ces backlogs ont pour particularité de mieux représenter les éléments de travail qui encapsulent les classiques User Stories tels que les **Features**. Cela permet, entre autres, de suivre le scénario dans lequel le management définit des Features au niveau d'un projet d'équipe, puis ces Features sont découpées en User Stories et réparties dans les différentes équipes qui composent ce projet d'équipe. On se retrouve ensuite dans un classique schéma de Product Backlog, qui répartira les User Stories dans des Sprints et les découpera en multiples tâches.

Les toutes nouvelles Team Rooms permettent de centraliser différents échanges. On peut y configurer des alertes telles que le résultat d'exécution d'une Build, la modification d'un Élément de travail, etc. et les résultats de ces alertes sont visibles par toute l'équipe directement dans la Team Room. La centralisation de ces communications possède plusieurs avantages :

- Elles deviennent un point de discussion de référence, particulièrement utile lorsque les membres d'une même équipe sont éloignés géographiquement
- Elles peuvent être consultées juste avant le Daily Meeting, afin que chaque membre de l'équipe puisse se remémorer rapidement ce qu'il a fait la veille
- Elles peuvent également servir de moyen de communication entre membres d'équipes différentes

Annoncée très récemment, une nouvelle section appelée **«Charts»** fait son apparition dans la partie Éléments de travail. Cette nouvelle option permet de configurer rapidement et simplement quelques **graphiques concernant les états des différents éléments de travail** du projet d'équipe. Ces graphiques sont ensuite disponibles par toute l'équipe depuis le Team Web Access et constituent un tableau de bord simple et efficace [Fig.5].

Enfin, pour finir, une nouvelle fonctionnalité qui laisse présager d'intéressants scénarios dans le futur : La possibilité de **commenter du code directement depuis l'interface web de TFS**. Cette nouvelle option offre déjà des cas d'usages très intéressants, ne serait-ce que la possibilité de faire des revues de code depuis n'importe quel smartphone permettant d'accéder à l'interface web de TFS

> Testing

Outre l'ajout du support des Coded UI Tests avec les applications du Windows Store, la partie Test de Visual Studio 2013 se voit également dotée d'une **meilleure gestion des cas de tests depuis l'interface web**.

Tout comme Microsoft Test Manager nous le permet, il est maintenant possible de créer directement depuis le Team Web Access des plans de tests, d'utiliser des étapes partagées, de rajouter des paramètres sur les étapes du cas de test.

Le Microsoft Test Runner version web permet quant à lui de dérouler un cas de test depuis n'importe quel appareil pouvant se connecter au site web, tout en bénéficiant de l'assistant de création de Bug pré-remplis.

Dernière nouveauté concernant la partie tests : Le support des **tests de charges lancés depuis Windows Azure**. Lors de la création d'un projet de test de charge depuis Visual Studio 2013, il est maintenant possible de cibler directement Team Foundation Service en lieu et place du classique contrôleur de test.

Les utilisateurs virtuels sont alors simulés depuis Windows Azure et la mise en place de l'infrastructure d'exécution des tests s'en retrouve simplifiée. La version Preview possède une limite concernant le nombre d'utilisateurs virtuels disponibles, limite qui devrait être modifiée dès le passage en version RTM de Visual Studio 2013

> Modeling Tools

En plus de l'outillage de modélisation UML déjà existant dans Visual Studio 2012, un nouvel outil appelé Code Map semble déjà être une option de choix lorsqu'il s'agit de

comprendre et/ou debugger une application dont le code n'est pas entièrement maîtrisé (Que ce soit dans des scénarios de maintenance d'une application existante, ou tout simplement lorsque l'application a été écrite par plusieurs intervenants).

Les options d'interaction avec la Code Map sont accessibles depuis l'éditeur de texte Visual Studio 2013. Il suffit par exemple d'effectuer un clic droit sur le nom d'une méthode pour pouvoir l'ajouter à une Code Map existante, ou même en créer une nouvelle.

Une fois que tous les éléments que l'on veut observer sont présents sur la Code Map, il est possible de demander à partir de cette dernière l'ajout automatique d'autres éléments tels que toutes les méthodes faisant appel à un objet particulier, etc. À chaque ajout, la Code Map dessine automatiquement les relations entre les différents objets, et il est alors plus facile de voir si les relations semblent cohérentes avec le fonctionnement attendu de l'application.

En cas de besoin, il est également possible d'annoter une Code Map, afin de fournir divers indices de compréhensions et/ou de repérage. On peut aussi sauvegarder et partager cette Code Map afin de demander l'avis d'un collaborateur la concernant, ou tout simplement pour conserver un historique.

Enfin, s'il est non seulement possible de suivre le débogage d'une application à travers la Code Map, il est également possible

d'en générer une en plein débogage, à partir des informations récoltées par la Call Stack. (Note : en plein débogage, les éléments parcourus par le débogage sont automatiquement ajoutés sur la Code Map)

CONCLUSION

Cette version 2013 de Visual Studio apporte bon nombre de nouveautés particulièrement intéressantes. Il est plaisant de constater que Microsoft fait un effort remarquable pour suivre un cycle de Release de plus en plus court, comme on a pu le constater au travers des nombreux updates de Visual Studio 2012 l'année dernière.

Cependant, Microsoft ne compte pas en rester là, et les différents chefs de divisions ont affirmé durant la Microsoft Build 2013 qu'ils comptaient dorénavant offrir la possibilité aux développeurs d'accéder à des versions du Framework .NET en avance de phase au travers d'un repository NuGet officiel (<http://www.nuget.org/profiles/dotnetframework/>) Il ne reste plus qu'à attendre la sortie officielle prévue le 13 novembre prochain, pendant laquelle on espère en savoir plus sur leur nouvel outil de déploiement, basé sur le rachat du produit InRelease.

Vivien Fabing
Consultant ALM

vfabing@infinitesquare.com
<http://blog.infinitesquare.com/b/vivien>



INFINITE SQUARE

Restez connecté(e) à l'actualité !

- **L'actu** de Programmez.com : le fil d'info **quotidien**
- La **newsletter hebdo** : la synthèse des informations indispensables.
- **Agenda** : Tous les salons, barcamp et conférences.

Abonnez-vous, c'est gratuit !

www.programmez.com

The screenshot shows the Programmez.com website with a navigation bar at the top. The main content area is divided into several sections: 'Abonnement' (Subscription), 'Concours Azure/Raytracer' (Contests), 'Nouveaux flashs sur Programmez!' (New flashes on Programmez!), 'Actes / News / Livres / Agenda' (Acts / News / Books / Agenda), 'SUPERMON for JAVA' (SUPERMON for JAVA), 'Microsoft' (Microsoft), 'Tribune Libre' (Free Tribune), and 'UN BUG « VISSEUX »' (A BUG « VISSEUX »). There are also links to 'Rechercher' (Search), 'RSS', 'NEWSLETTER', and 'CONTACT'.

Java 8 : le tour du propriétaire

JDK 8 ou Java 8 est le sujet de discussion des programmeurs Java sur de nombreux forums. On y détaille ses avantages et inconvénients, on se prépare à son utilisation. Certains la considèrent comme la version la plus importante depuis Java 1. Fedora 19 par exemple, arrivée en juillet 2013, contient deux versions de OpenJDK pour Java 7 et 8. Il est à signaler que l'amélioration touche l'ensemble de la plate-forme Java : JVM, classes et API. Il est vrai aussi, que la partie la plus importante des nouveautés, tend à transformer Java en un langage fonctionnel.

Java 8 est toujours en développement, mais une version «Early Access» avec des Builds continues, est disponible à l'URL suivante : <https://jdk8.java.net/download.html>

Acceptez la licence et sélectionnez une version de JDK pour votre OS. Par exemple un fichier .gz pour une distribution Linux 64 bits.

> Installation

Décompressez le fichier .gz. Vous aurez un dossier jdk1.8.0, les fichiers binaires sont dans le sous-dossier « bin »

Note : Si vous comptez utiliser régulièrement les fichiers binaires ou exécutables dans le dossier « bin » de Java 8, il est conseillé d'ajouter ce dossier au chemin des exécutables de votre système (path). Pour le faire dans Linux par exemple, éditez le fichier .bashrc

> Quel IDE ?

Netbeans 7.4 RC1 (<https://netbeans.org/>), ou plus, supporte beaucoup de nouveautés de Java 8, out-of-the-box et constitue la plate-forme de choix pour pouvoir la tester. Malheureusement IntelliJ IDEA et Eclipse sont pour l'instant en retard. Une avant-version pour IntelliJ IDEA (EAP or Early Access Program), ainsi qu'une version non-officielle d'Eclipse sont malgré tout disponibles, si on ne peut pas se séparer de son IDE préféré pour les tests de Java 8. Pour cet article je vais utiliser NetBeans 7.4 RC2.

> Préparer NetBeans pour Java 8

Après avoir installé Java 8 et NetBeans. Commencez par créer un nouveau projet Java > puis un clic droit sur le projet nouvellement créé > **Properties**.

Sous **Categories** (à gauche) > un clic sur **Libraries** > puis un clic sur le bouton **Manage Platforms**.

Dans le nouveau dialogue > Cliquez sur le bouton **Add Platform...** Sélectionnez le dossier obtenu en décompressant ou installant Java 8. Donnez un nom à cette plateforme, puis Finish. Cliquez sur **Close** pour retourner à la boîte principale de Properties.

Un clic sur **Sources** (sous Categories) > sélectionnez Java 8 (ou le nom que vous avez donné à la plateforme) dans la liste **Source/Binary Format**. Pour terminer, un clic sur OK.

> Lambda Expressions

Vous connaissez les classes anonymes en Java ? Vous avez bien aimé leur utilisation ? Java 8 vous propose d'aller encore un peu plus en avant avec les méthodes anonymes !!!

> Un petit rappel des classes anonymes (Anonymous inner classes)

Ce sont des classes que l'on peut déclarer pour obtenir un objet en

même temps. Prenons l'exemple d'un bouton Swing qu'on désire avoir avec un comportement particulier lorsque l'utilisateur le clique. Pour ce faire nous ajoutons au bouton un ActionListener, qui écoute les événements de clics sur un bouton et invoque une méthode qui répond au clic. La classe passée à la méthode addActionListener du bouton suivant est une classe anonyme qui implémente l'interface ActionListener. Elle est anonyme, sans nom. On ne lui assigne pas de nom, car on a besoin de l'utiliser une seule fois ici. Il y a bien sûr des classes internes utilisées pour répondre à plusieurs composants et qui ont nécessairement besoin d'un nom, mais ce n'est pas le cas ici. La classe passée à la méthode addActionListener est utilisée une seule et unique fois.

```

jButton1.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent event)
    {
        System.out.println("Hello World!!!");
    }
});

```

Remarquez la syntaxe : on déclare une classe après le mot clé «new» qui implémente une interface et on passe comme argument la méthode. Il y a aussi des classes anonymes qui étendent une autre classe et passent ou non des arguments au constructeur. Notez au passage que les classes anonymes sont des expressions.

> Interface fonctionnelle

Ce sont des interfaces qui disposent d'une seule méthode (seule méthode abstract, plus de détails dans amélioration des interfaces), exactement comme l'interface ActionListener et sont des candidats à l'utilisation des expressions lambda avec des méthodes anonymes. Prenons l'exemple ActionListener. Avec une expression Lambda on va dire que la méthode addActionListener reçoit une interface qui dispose d'une méthode unique, donc pas besoin de donner le nom de cette méthode et ceci donne l'expression suivante :

```

jButton1.addActionListener((ActionEvent evt) -> {
    System.out.println("Hello World!!!");
});

```

Au départ, c'est vraiment déroutant, mais si on y réfléchit un peu, il est tout à fait logique que la méthode addActionListener prenne une ActionListener qui a une seule méthode. Au lieu d'encombrer le code, on passe le paramètre et on demande le traitement après le signe «->». C'est tout. Une petite remarque au passage, le compilateur se charge de tout, en arrière-plan, pour créer la classe anonyme.

me et le reste, pour nous donner cette lisibilité de code.
Même logique, pourquoi indiquer le type de l'argument ou paramètre passé à la méthode, étant donné que cette méthode ne prend qu'un seul paramètre du type ActionListener
Supprimer le type du paramètre donne :

```
jButton1.addActionListener( (evt) -> {
    System.out.println("Hello World!!!");
});
```

Bien entendu, vous donnez le nom de votre choix au paramètre.
Et si on supprime les parenthèses, qui ne servent plus à rien, on arrive enfin à cette expression :

```
jButton1.addActionListener(e -> {
    System.out.println("Hello World!!!");
});
```

> Type Inference

Quand on utilise une expression Lambda, le compilateur se charge de détecter à la fois le type du paramètre(s) et du résultat obtenu.
Nous avons une classe Personne et nous voulons trier une liste de cette classe. Nous pouvons par exemple utiliser Collections.sort(), en passant une liste et une classe anonyme, qui implémente Comparator comme nous avons l'habitude de le faire, ou sous forme d'une expression Lambda comme suite :

```
package com.java_javafx.programmez_article_java8;

import java.util.*;

/**
 *
 * @author Kaesar ALNIJRES
 */
public class Personne {
    private String nom;
    private int age;

    public Personne(String nom,int age)
    {
        this.nom=nom;
        this.age=age;
    }

    public String getNom() {
        return nom;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }
}
```

```
public int getAge() {
    return age;
}

public void setAge(int age) {
    this.age = age;
}

public String toString()
{
    return «»+nom+»:»+age;
}

public static void main(String[] args)
{
    List<Personne> list=new ArrayList<Personne>();

    list.add(new Personne("vingt",20));
    list.add(new Personne("cinquante",50));
    list.add(new Personne("soixante",60));

    //trier en utilisant une class anonyme
    Collections.sort(list, new Comparator<Personne>()
        {

            public int compare(Personne p1, Personne p2)
            {
                return p1.getAge() - p2.getAge();
            }
        }
    );

    //afficher la liste triée
    for(Personne ps:list)
        System.out.println(ps);

    System.out.println("=====");

    //Exactement la même chose, en utilisant une expression Lambda
    Collections.sort(list, (p1, p2) -> p1.getAge() - p2.getAge());

    //afficher la liste triée
    for(Personne ps:list)
        System.out.println(ps);

    System.out.println("=====");
}}
```

Vous préférez ça ?

```
Collections.sort(list, new Comparator<Personne>()
{
}
```

```
public int compare(Personne p1, Personne p2)
{
    return p1.getAge() - p2.getAge();
}
}
```

Ou ça

```
Collections.sort(list, (p1, p2) -> p1.getAge() - p2.getAge());
```

Et si vous voulez trier selon le nom au lieu de l'âge ?

```
Collections.sort(list, (p1, p2) -> p1.getNom().compareTo(p2.getNom()));
```

La syntaxe :

- Un ou plusieurs paramètres de la méthode anonyme, avec leurs types séparés par une virgule « , » entre parenthèses. Si un paramètre unique, plus besoin de son type ou des parenthèses.
- Le signe « -> »
- Un corps, qui contient une seule expression ou un bloc d'instructions.

Note : Quand on utilise une seule expression, La JVM évalue l'expression et retourne sa valeur. On ne peut utiliser une instruction « return » qu'entre des accolades, car « return » n'est pas une expression.

```
Collections.sort(list, (p1, p2) -> {
    return (p1.getAge() - p2.getAge());
});
```

Accès aux autres membres :

- Une expression Lambda peut accéder aux autres variables et méthodes globales de sa classe, y compris privées, exactement comme les classes internes.
- Elle peut aussi accéder aux variables locales, comme les classes locales et anonymes, si ces variables sont finales ou considérées comme telles (final or effectively final). Effectively final veut dire que les variables n'ont pas changé de valeur depuis leurs initialisations.
- Une expression Lambda est sans effet de Shadowing là où une variable cache une variable d'une portée plus globale portant le même nom. Par exemple, on ne peut pas utiliser le nom d'une variable déclarée dans une méthode en tant que paramètre d'une expression Lambda, ni d'ailleurs le paramètre formel passé à la méthode contenant cette expression.

> Amélioration sur les Interfaces

Defender Methods :

Les interfaces acceptent maintenant la définition des méthodes et notamment des « defender methods », ou bien les méthodes par défaut. Une méthode par défaut fournit à l'interface son comportement par défaut. Les defender methods, sont très utiles pour l'utilisation des expressions Lambda. Pour définir une méthode par défaut, on utilise le mot clé **default**. Exemple : dans le nouveau JDK, l'interface Iterable implémente une méthode default. Quelques autres interfaces ont aussi des méthodes par défaut dans le JDK8 :

```
default void forEach(Consumer<? super T> action) {
```

```
Objects.requireNonNull(action);
for (T t : this) {
    action.accept(t);
}
}
```

Enfin, un petit rappel sur les méthodes génériques.

> Méthodes de type générique

La déclaration d'une méthode peut être générique, c'est ce qui rend le traitement de plusieurs types d'objets transparent pour l'utilisateur. Si vous voulez par exemple coder une méthode capable de prendre un array (ou tableau) de n'importe quel type d'objet et retourner une collection remplie d'éléments de ce même type. Il est facile d'utiliser les méthodes génériques. Voici un exemple de la méthode arrayToCollection : code complet sur www.programmez.com La méthode va prendre un tableau de type générique, construire une variable de Collection également de type générique, et retourner une Collection du même type. L'avantage est qu'on peut maintenant utiliser cette méthode pour tous types d'objets. On n'a pas besoin de passer le type d'argument à la méthode générique. Le compilateur se charge de trouver le type (Code sur www.programmez.com).

Un nouveau package contenant des interfaces fonctionnelles très utiles (Generic Interfaces) a été ajouté par l'équipe de Java. Par exemple l'interface Predicate<T> est utilisée pour effectuer des tests sur des objets d'une classe donnée. Je cite ici deux interfaces et je vous incite à consulter les documentations.

L'interface Predicate<T>

Elle prend un type T et retourne une boolean (T -> boolean).

```
interface Predicate<T> {
    boolean test(T t);
}
```

L'interface Predicate a une seule méthode, qui prend un objet de même type, et retourne true ou false selon la condition appliquée.

Exemple :

Dans la même classe Personne, on peut imaginer une méthode pour tester si une ou plusieurs personnes remplissent des conditions en utilisant l'interface Predicate<T>

```
package com.java_javafx.programmez_article_java8;
import java.util.*;
import java.util.function.Predicate;

/**
 *
 * @author Kaesar ALNIJRES
 */
public class Personne {
    ...
    ...
    public static boolean testAge(Personne p, Predicate<Personne> predicate)
    {
        return predicate.test(p);
    }
}
```



```

}

...

...

public static void main(String[] args)
{
    List<Personne> list=new ArrayList<Personne>();

    list.add(new Personne(«vingt»,20));
    list.add(new Personne(«cinquante»,50));
    list.add(new Personne(«soixante»,60));

    System.out.println(«Afficher uniquement > 30»);
    for(Personne ps:list)
        //Une expression Lambda est utilisée pour tester si la personne
        a dépassé 30 ans
        System.out.println(ps+» «+testAge(ps,p ->p.age >30 ));
}

```

> L'interface Consumer<T>

Elle prend un type T et produit une action sur le type (T -> void). Elle a une seule méthode accept dans laquelle on applique l'action :

```

public interface Consumer<T>
{
    void accept(T t) ;
}

```

Exemple :

Si on désire dans la classe Personne, ajouter 10 à l'âge d'une personne.

```
Consumer<Personne> consum=p -> p.setAge(p.getAge()+10);
```

Note : Il y a bien sûr d'autres interfaces dans ce package. Je vous conseille vivement d'y jeter un œil, tout en gardant à l'esprit que Java 8 est toujours en développement; des changements ne sont donc pas à exclure.

> Java.util.stream

Nouveau type en Java. Il représente une séquence d'éléments (supporte aussi le travail en parallèle), permettant de chaîner des opérations et agrégations sur les éléments qui le composent. L'avantage est d'avoir une série d'opérations sur des éléments à la manière de pipes d'Unix où le résultat d'une opération est le sujet d'une autre opération, etc. Voyez les streams comme une enveloppe qui entoure les données afin d'effectuer des suites d'opérations, comme le filtrage, l'exécution, ou tout autre pipeline d'opérations.

Les Streams ne sont pas donc considérés comme d'autres structures de données (data structure). Pas d'organisation interne d'éléments et pas de stockage. La plupart des méthodes de Stream prennent un stream et produisent un stream en sortie, sauf les méthodes finales où le stream est consommé.

Un stream permet aussi le travail en multithreading avec un minimum de code et utilise le principe de consommation d'entrées. Enfin, les streams sont désignés spécialement pour Lambda [Fig.6].

> Obtenir un stream

Il existe plusieurs moyens d'obtenir un stream. Deux méthodes statiques «Stream.of()» par exemple permettent d'avoir des streams d'une valeur ou d'un tableau d'éléments (array). De même il est pos-

sible d'appeler la méthode stream(), à la fin d'une collection, comme une liste pour avoir son stream. Si je prends l'exemple de la classe Personne. Je peux tout simplement obtenir un stream à partir de la liste des Personnes en appelant la méthode stream(). A partir de ce moment je peux appliquer des méthodes de l'interface Stream pour enchaîner les opérations sur des éléments.

```

List<Personne> list=new ArrayList<Personne>();
list.add(new Personne(«vingt»,20));
list.add(new Personne(«cinquante»,50));
list.add(new Personne(«soixante»,60));
list.stream();

```

ou

```
static <T> Stream<T> of(T t)
```

ou

```
@SafeVarargs
static <T> Stream<T> of(T... values)
```

Ou

```
Integer[] a={5,3,7};
Stream.of(a);
```

etc.

> Quelques méthodes

Stream dispose de beaucoup de méthodes qui utilisent des interfaces fonctionnelles vues précédemment pour effectuer ses pipeline d'opérations. Je prendrai comme exemple, filter(), map() et forEach().

Map()

```
<R> Stream<R> map(Function<? super T,? extends R> mapper)
```

C'est une opération intermédiaire (stream entrée, stream sortie) Prend un stream, applique sur chaque élément une fonction, et retourne un nouveau stream. Par exemple sur une liste de Personne, je veux uniquement la partie âge ou l'e-mail.

ForEach()

```
void forEach(Consumer<? super T> action)
```

C'est une opération terminale (stream entrée). Le stream est totalement consommé après cette méthode. Elle applique une action sur chaque élément dans le stream.

filter()

```
Stream<T> filter(Predicate<? super T> predicate)
```

C'est une opération intermédiaire (stream entrée, stream sortie) Prend un stream et retourne un nouveaux stream contenant uniquement les éléments ayant passé le test de Predicate.

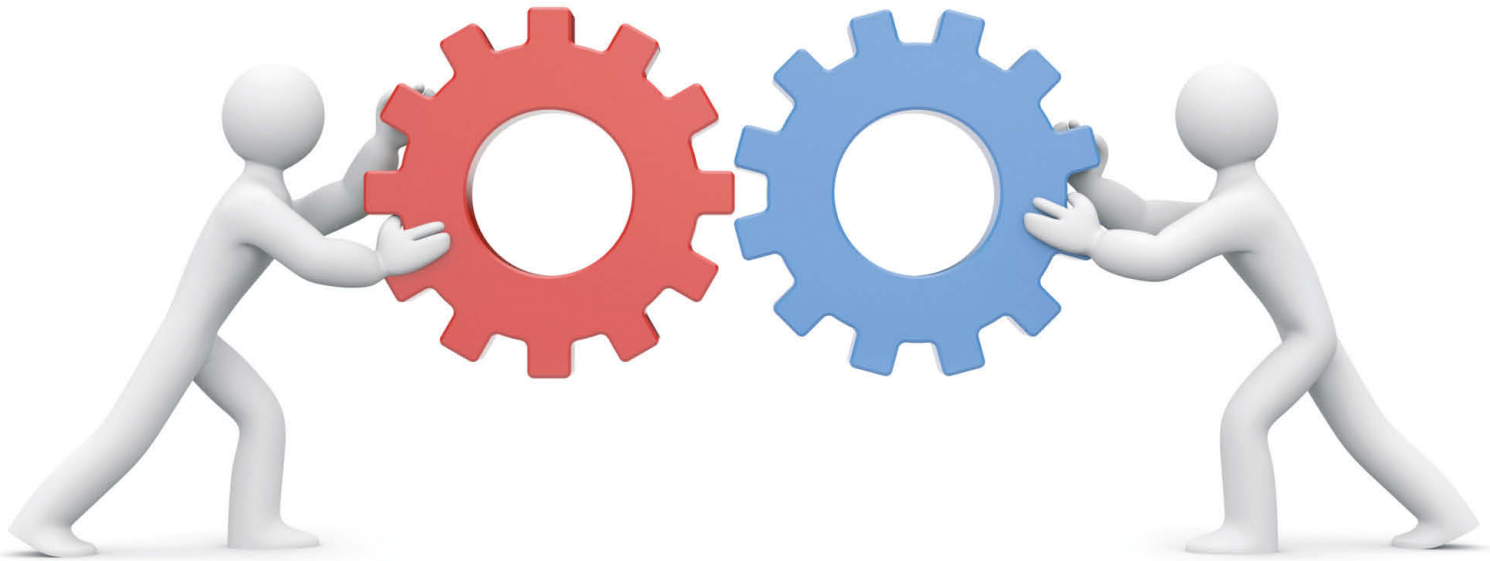
> Pour finir

Il y a beaucoup de choses dans Java 8. Il est difficile, voire impossible de mentionner toutes les nouveautés en un seul article. Nous y reviendrons prochainement. Bonne découverte !

Kaesar Alnires - Développeur Java
www.java-javafx.com



DevOps: « non ce n'est pas du marketing, c'est une révolution ! »



Vous l'aurez remarqué, DevOps est le mot à la mode en ce moment. Tout le monde en fait ou prétend en faire mais en général, nous sommes bien loin du véritable DevOps. Martin Nally (IBM Fellow, ancien directeur technique d'IBM Rational), durant la conférence Technical Summit IBM le 17 octobre dernier, a expliqué comment et pourquoi une véritable rupture a lieu grâce (ou à cause) aux Google, Facebook, Twitter, Netflix, etc.

Une des conclusions de Martin est la mort du concept de « versions » (la fameuse release). On délivre en continu et tout le temps : Facebook fait des dizaines de corrections et de modifications par jour, ces acteurs n'hésitent pas à changer de modèles : Itsy c'est plus de 6000 changements et corrections de codes en 2012 ! Mais l'autre force est la capacité à n'avoir qu'un code à gérer et à pouvoir le tester en production sur un groupe d'utilisateurs définis.

C'est un changement philosophique radical. La notion de « release » est le standard pour de nom-

breuses entreprises et les développeurs sont habitués à cela : des versions, une ou plusieurs par an, cycle long. Les méthodes agiles avaient déjà bousculé cela en imposant des cycles courts.

Mais un problème se pose : l'ADN ! Transformer un développeur, une équipe, à la culture du « cycle sans fin », est un Everest. La rupture peut être violente car c'est remettre en cause 30 ans de méthodes et d'outils. Martin pense, à juste titre, qu'il faut créer de nouvelles équipes ayant cet ADN...

Je vous rassure, DevOps ne préconise pas de tout casser. Mais rapprocher le développeur et la production n'est pas chose facile car ce sont parfois deux mondes très différents. Le déploiement continu modifie les habitudes et les rythmes des développements, au risque de fournir des applications plus instables et d'une qualité incertaine.

illustration : tiré du livre blanc « the invisible thread » (IBM)

<http://ibm.co/OvkXdB>

François Tonic

Mettre en place une culture DevOps

DevOps est le terme hype du moment dont on entend beaucoup parler. Ce sont des compétences que les entreprises recherchent sur votre CV. Elles s'arrachent aujourd'hui les profils ayant une expérience dans la mise en place des pratiques DevOps. Mais qu'y a-t-il de réellement caché derrière ce terme ?

> Définition

Le terme DevOps vient de la contraction des deux termes anglais : Development qui désigne les équipes de développement et Operations qui correspond à l'équipe en charge de l'exploitation des serveurs (les administrateurs systèmes et réseaux) [Fig.1]. DevOps est en fait un mouvement récent visant à rapprocher ces deux équipes et à aligner leurs objectifs sur les besoins de l'entreprise. Pour bien comprendre ce mouvement, il faut d'abord en comprendre les causes de son apparition.

> Deux mondes qui ont l'habitude de s'affronter : les développeurs contre l'exploitation

Quel développeur n'a jamais entendu qu'il faudra attendre plusieurs semaines, voire des mois, avant de pouvoir mettre en production son application ? Quel administrateur système n'a jamais eu de problème en déployant une application en production et entendu la phrase mythique du développeur : « Pourtant ça marche sur mon poste » ?

> Mais pourquoi tant de haine ?

On ne peut pas dire que les relations entre les équipes de développement et l'exploitation soient des plus chaleureuses. Généralement, c'est même plutôt l'inverse. Rares sont les entreprises où ces équipes s'enten-

dent bien et communiquent régulièrement. Plusieurs raisons à cette situation.

Cloisonnement et organisation en silo

La plupart des services informatiques sont organisés en silo. Les équipes de développement, celles de la qualité, celles en charge de l'exploitation et celles qui s'occupent du support applicatif, sont séparées dans plusieurs services distincts.

Ces différents services ne se parlent généralement pas ou uniquement lors des mises en production, ou quand un problème survient. Et bien entendu, ce n'est pas forcément le même interlocuteur à chaque fois. Ce type d'organisation ne facilite pas les échanges et la collaboration entre les équipes. Et pour couronner le tout, ces services sont éloignés physiquement. Les développeurs sont proches du métier et l'exploitation proche des serveurs. Les équipes ne se rencontrent presque jamais physiquement.

Éloignement culturel

En plus d'être cloisonnés, ces deux mondes sont totalement éloignés culturellement et ont des objectifs diamétralement opposés. Les développeurs cherchent à répondre rapidement aux besoins et demandes d'évolutions du métier. Pour y parvenir, ils s'appuient sur les méthodes agiles. Cela se traduit par des livraisons et des demandes de mise en production beaucoup plus rap-

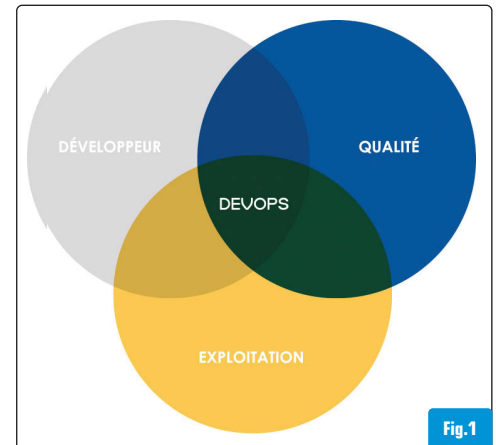


Fig.1

prochées. De l'autre côté, l'équipe d'exploitation doit assurer au système d'information de l'entreprise un environnement stable, sécurisé et pérenne dans le temps. Pour répondre à ces contraintes, le service est organisé suivant les recommandations ITIL.

Le problème de cette organisation c'est qu'elle se traduit par une certaine lourdeur administrative : beaucoup de documents ou d'outils à renseigner pour une mise en production, des validations par un comité lors des demandes de changement ou d'évolutions... Du coup les délais de mise en production deviennent excessivement longs. Il faut plusieurs semaines voire mois, avant d'avoir des ressources matérielles nécessaires ou qu'un administrateur système puisse s'occuper de la mise en production [Fig.2].

Environnement de développement différent de celui de production

Il n'est pas toujours possible pour des raisons de coût et de mobilisation de ressources d'avoir des environnements (test, dev, recette, production) tous identiques. Sur des infrastructures complexes, les développeurs ne peuvent, au mieux, tester leur application dans un environnement proche de la production qu'à partir de la recette. Avant d'arriver à cet environnement ils testent son application dans un environnement très différent avec :

- OS différents
- Serveurs et versions différentes
- Configuration différente : pas de cluster ni de load-balancer
- Règles de sécurité différentes

Ces différences se traduisent généralement par des mises en recette ou en production douloureuses.

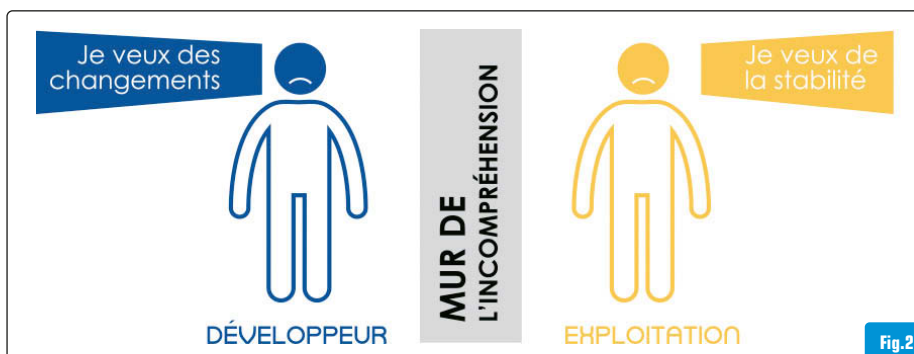


Fig.2



> Les pratiques et outils DevOps

Comme pour les méthodes agiles, qui ont rapproché les développeurs et le métier, le mouvement DevOps vise à rapprocher les équipes de développement et d'exploitation [Fig.3]. Pour cela il s'appuie sur plusieurs processus, des bonnes pratiques et des outils.

> Les pratiques DevOps Organisation

L'organisation est le sujet le plus complexe et sensible à mettre en œuvre. Mais c'est un prérequis fondamental pour réussir la mise en place des pratiques DevOps. Tout d'abord il faut casser les silos pour fluidifier la communication entre les équipes. Et quoi de mieux pour faciliter cette communication que de regrouper les deux disciplines au sein d'une même équipe et de les rapprocher physiquement. Idéalement dans un même bureau. Au-delà de partager un même bureau, il est intéressant de partager les mêmes outils. Il faut que les développeurs et l'exploitation comprennent les contraintes de chacun, et qu'ils recherchent ensemble de façon constructive des solutions à leurs problèmes. L'équipe doit avoir un objectif commun et non des objectifs différents et opposés. Sans quoi les bénéfices des pratiques DevOps seront vite limités.

Continuous Integration

L'intégration continue consiste à compiler, tester et livrer une application régulièrement. C'est un sujet qui est normalement aujourd'hui bien maîtrisé par les équipes de développement.

En résumé, les pratiques de base pour faire de l'intégration continue sont :

- Gérer dans un repository le code source type SVN, Git, Mercurial. Une bonne pratique est de gérer en parallèle des Branches par demande d'évolution ou versions
- Compiler et construire l'application de façon automatique

- Écrire des tests unitaires et les exécuter régulièrement ou lors de la compilation
- Construire et packager l'application à intervalle régulier (toutes les nuits) ou déclencher la construction après chaque commit. Et informer l'équipe de développement si un problème apparaît

Continuous Delivery

Le Continuous Delivery ou Livraison continue, est une pratique visant à automatiser le déploiement d'une application sur différents environnements tout au long de son cycle de vie : Dev, Integration, Recette, Production.

Pour y parvenir il est nécessaire d'avoir une application correctement packagée et que les différents environnements soient le plus proche possible de celui de production. L'application doit pouvoir être déployée sur les différents environnements sans être modifiée. Exit donc les fichiers de configuration spécifiques à chaque environnement nécessitant une recompilation de l'application.

Même si le déploiement est 100 % automatisé, la mise en production peut nécessiter une validation manuelle.

Continuous Deployment

Le Continuous Deployment est la pratique qui vient après celle du Continuous Delivery. La différence principale avec le Continuous Delivery c'est que les livrables sont déployés automatiquement jusqu'en production. Les géants du web font cela plusieurs fois par jour sans que leurs utilisateurs ne s'en rendent compte. C'est la pratique que je considère comme le Saint Graal !

Le Continuous Delivery doit d'abord être mis en œuvre avant de pouvoir faire du Continuous Deployment. Si le Continuous Delivery peut être mis en œuvre simplement et rapidement pour toutes les applications, le Continuous Deployment est difficilement applicable tout le temps. Cette pratique nécessite d'avoir une grande confiance dans les modifications qui sont envoyées en pro-

duction. Il faut pour cela réussir à automatiser les tests d'intégration.

Infrastructure as Code

L'Infrastructure as Code consiste à définir son infrastructure sous forme de code.

Pour assurer une plus grande stabilité, on applique les mêmes pratiques que pour le développement à l'infrastructure :

- Tests unitaires
- Intégration continue

L'objectif est d'avoir un environnement identique du dev à la production.

En dev, on utilise des VM lancées sur le poste du développeur. On y installe les mêmes outils que pour la production.

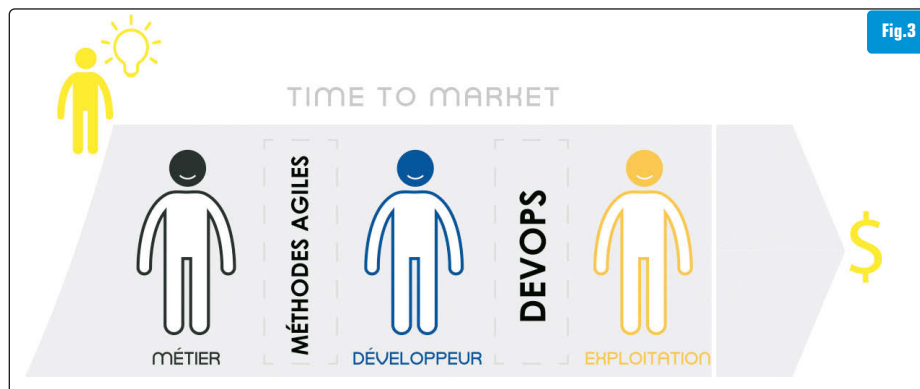
Amélioration continue

Tout automatiser est une bonne pratique, mais automatiser et répéter inlassablement les mêmes erreurs n'en est définitivement pas une. La culture DevOps implique d'améliorer continuellement ses processus et ses outils. La première étape consiste à récolter régulièrement des feedbacks et des indicateurs. Puis sur la base de ces informations, formaliser, définir puis suivre un plan d'amélioration. Attention toutefois à ne pas prendre des indicateurs qui risqueraient de faire entrer en conflit les développeurs et l'exploitation. Les indicateurs à prendre en compte doivent être ceux permettant de réduire le Time to Market.

Les autres processus

Même si on associe très souvent DevOps au Continuous Deployment, toutes les applications ont un cycle de vie qui ne revient pas exclusivement à mettre en production rapidement et en continu de nouvelles versions. Tout au long de ce cycle de vie il va falloir gérer :

- Le monitoring et les logs pour remonter des informations pertinentes afin de faciliter le diagnostic et la résolution de problèmes,
- Les incidents et problèmes en réunissant dans une même pièce les développeurs et l'exploitation et en donnant aux développeurs un accès aux logs, et si possible un accès en lecture seule aux serveurs,
- Les performances avec une intégration en continu de tests de performance, en dimensionnant correctement son infrastructure et en automatisant la montée en charge (autoscaling),
- La sécurité en intégrant la vérification automatique des failles de sécurité,



- La continuité de service et le plan de reprise d'activité.

> Les outils

La part des outils est très importante dans la culture DevOps. Cependant il n'existe pas un unique outil magique qui permette de tout faire [Fig.4]. Il est donc très fréquent de prendre plusieurs outils et de les assembler entre eux pour arriver à ses fins. Et ce, même si ces outils ont des recouvrements de fonctionnalités.

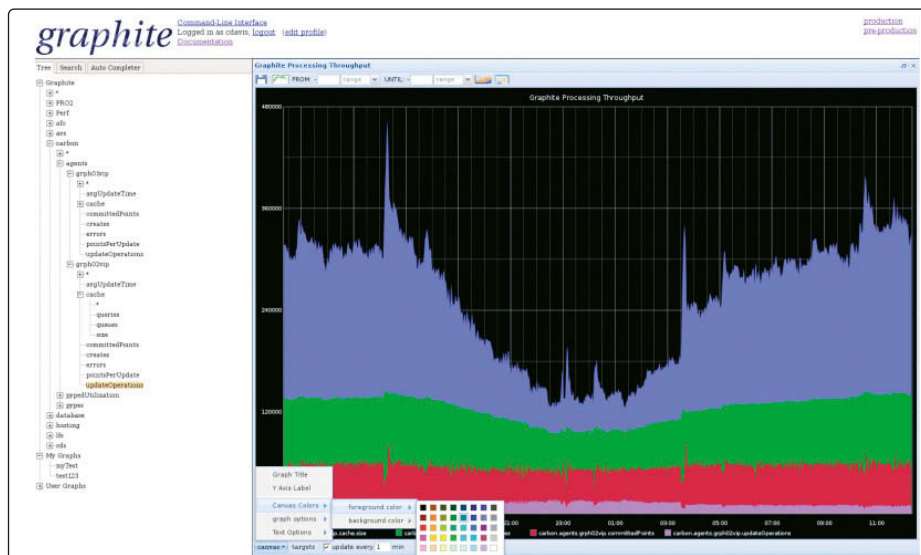
> Quels outils pour quels besoins :

Gestion des versions

Les outils de gestion de version tels que SVN, Git, Mercurial sont utilisés aussi bien par les développeurs pour gérer leur code que par l'exploitation pour gérer leurs scripts et le code définissant l'infrastructure. Quel que soit l'outil que vous utilisez, il est important de définir un workflow pour la gestion des branches et des tags.

Outils de build et de packaging

Non le livrable que vous mettez en production ne doit jamais être le résultat de la compilation des fichiers par votre IDE favori ! Donc quel que soit le langage que vous utilisez pour développer votre application, appuyez-vous sur un outil adapté à celui-ci. Utilisez des Maven, Ant, Ivy, Gradle, Rake, ou équivalents pour packager et livrer votre application. Celle-ci peut être livrée dans le format de votre choix : par exemple, fichier ZIP, JAR, WAR, package Gem, Egg, RPM, DEB, Tag dans un repository Git... Ce qui compte c'est que vous puissiez la déployer simplement et sans modification dans les différents environnements. Le format de livraison doit être défini avec votre exploitation pour faciliter son déploiement en automatique. Pour une révision donnée, votre application doit pouvoir impérativement être recompilée à partir des sources, et produire le même livrable. Les fichiers de configuration nécessaires à votre application doivent être externalisés. Les outils de gestion de configuration vont se charger de déployer et paramétrer ces fichiers pour chaque environnement. Les livrables peuvent être envoyés dans un repository adapté au for-



Name	Last Success	Last Failure	Last Duration	LC
gerrit-master	5 mo 6 days - #4011	3 mo 14 days - #4252	10 min	✓
lib-jenkins-maven-artifact-manager	2 mo 12 days - #1356	10 mo - #1352	54 sec	✓
lib-jenkins-maven-embedder	15 days - #1382	10 mo - #1370	3 min 34 sec	✓
lib-lira-api	10 days - #5350	2 mo 18 days - #5346	38 sec	✓
model-ant-project	1 yr 8 mo - #3	N/A	18 sec	✓

mat du livrable (Nexus, Archiva, Debarchiver, Reprepro, Gem In A Box...).

Intégration continue

Une fois que vous êtes capable de compiler et de générer un livrable via un outil de build, une bonne pratique consiste à le construire à intervalle régulier ou en fonction d'un événement (par exemple après chaque commit). Les outils phares sont : Jenkins, Hudson, Travis. Vous pouvez également activer des outils d'analyse de la qualité de votre code (SonarQube par exemple) et essayer d'appliquer les recommandations faites par ces outils pour l'améliorer au fil des versions.

Infrastructure Virtuelle

Comme nous l'avons vu, il est important de pouvoir tester rapidement son application dans un environnement proche de celui de la production. On ne peut pas toujours avoir plusieurs environnements identiques, car

cela monopoliserait trop de ressources pour des usages ponctuels.

Une des solutions consiste à mettre en place, à la demande, une infrastructure virtuelle proche de la production. L'outil Vagrant permet de décrire cette infrastructure et de la déployer. Il s'appuie sur la solution de virtualisation VirtualBox et pilote la création et le paramétrage des serveurs virtuels. En partant d'un simple fichier de configuration et en quelques minutes, on se retrouve avec une infrastructure pouvant être composée de plusieurs serveurs. Il est facile de décrire une architecture type : Frontal Web, 2 serveurs d'applications, un serveur pour la base de données.

Provisioning des serveurs et installation des systèmes d'exploitation

L'automatisation va jusqu'à l'installation en automatique et par le réseau des systèmes

Fig.4

INSTALLATION/
PROVISIONING

CONFIGURER

DÉVELOPPEMENT

PACHAGER

DÉPLOYER

MONITORER



d'exploitation sur des serveurs physiques. Les solutions Razor et Crowbar s'occupent de piloter à distance ces installations. Pour les serveurs virtuels, les outils Veeva et Packer.io vont automatiser la création d'images virtuelles de base.

Gestion de la configuration

Une fois l'OS installé, les outils comme : Chef, Puppet, Cfengine, SaltStack, Ansible, permettent de configurer et d'installer les services nécessaires sur les serveurs. Ces outils utilisent des modules décrivant l'état dans lequel doit se trouver le serveur et appliquent les modifications nécessaires pour y parvenir. Ils fonctionnent généralement en mode agent/serveur et vont chercher la configuration à appliquer sur un serveur central.

Ils permettent de décrire les packages système à installer, les fichiers de configuration à modifier, les services à lancer ou arrêter, des utilisateurs à créer, des commandes à exécuter... Cette description est généralement regroupée dans un module par service. Par exemple un module pour Apache, un autre pour MySQL.

Tous ces outils proposent des mécanismes pour chercher et installer des modules développés et maintenus par la communauté. Beaucoup de modules existent.

Alors avant de développer le vôtre, regardez s'il n'en existe pas un qui répondrait à vos besoins.

Orchestration et déploiement applicatif

Les outils de gestion de configuration ne proposent pas nativement de solution d'orchestration. Ces outils ne permettent pas ce type de procédure :

- Arrêter les serveurs d'applications
- Lancer des scripts SQL pour la mise à jour de la base de données
- Redémarrer les serveurs d'applications
- Déployer l'application sur plusieurs serveurs
- Reconfigurer un frontal web pour pointer sur la nouvelle application.

C'est dans ces cas que sont utilisés RunDeck, ControlTier, Capistrano, Fabric. Une autre catégorie d'outils tels que Func et mCollective permet d'exécuter des actions en masses sur les serveurs. Par exemple : redémarrer un service, supprimer un fichier, etc.

Monitoring

Les outils actuels de monitoring ne sont pas



adaptés aux nouveaux besoins imposés par les infrastructures virtuelles et le cloud. Le courant #MonitoringSucks est apparu en 2011, critiquant le côté monolithique, complexe à configurer, difficilement scalable de ces outils (nagios en particulier). Il propose des solutions en s'appuyant non pas sur une solution unique, mais sur un assemblage de plusieurs outils facilement configurables et mettant à disposition des API pour les alimenter.

On retrouve des outils existants et des nouveaux. On peut citer :

- Collectd, Stats, JMXTrans, Metrics, Esper, Ganglia, Graphite, Cube, OpenTSDB pour la collecte, le stockage, le traitement et la représentation de données.
- Graylog2, Flume, Logstash, Elasticsearch + Kibana pour la gestion et le traitement des logs.

> Quand commencer à utiliser les pratiques et outils DevOps

Vous l'avez bien compris l'utilisation des pratiques DevOps implique un certain investissement. Il faut apprendre et monter en compétence sur de nouveaux outils. Donc tous les projets ne peuvent pas s'y prêter.

Alors sur quels type de projets peut-on commencer à mettre en place DevOps ? Les cas suivants seront propices :

- Si votre entreprise développe plusieurs applications avec un socle technique relativement proche
- Si vous développez une application qui est

pérenne dans le temps et qui évolue fréquemment

- Si vous êtes un éditeur de solution type SaaS ou une Startup avec un service devant "scaler" à la demande

Pour des projets plus petits ou avec un budget restreint, il faudra étudier au cas par cas et analyser le retour sur investissement. Dans tous les cas, pour commencer vous n'êtes pas obligé de tout automatiser en une seule fois. Il est recommandé de commencer petit et d'automatiser régulièrement et petit à petit votre infrastructure.

> Mettre en place DevOps dans votre société

Comme beaucoup de changements dans une entreprise, la culture et l'organisation DevOps seront difficiles à mettre en œuvre sans appui et implication de sa direction. Mais tout n'est pas perdu, vous pouvez commencer à votre niveau.

Pour les développeurs vous pouvez commencer par utiliser un outil de build et mettre en place des tests unitaires de votre application. Gérez bien votre code source dans un repository, et faites des builds continus.

Un des meilleurs investissements que vous puissiez faire pour votre carrière c'est d'apprendre et d'être à l'aise avec un shell. Commencez par exemple par installer un Linux chez vous et à l'utiliser. À moins d'être dans un environnement 100 % Microsoft, cela vous servira toujours. Vous comprendrez également mieux l'environnement dans lequel les Ops évoluent.

Pour les Ops : commencez à automatiser votre infrastructure. Rédiger une procédure manuelle d'installation ne revient pas à faire de l'automatisation. Mais cette procédure vous servira de base pour essayer de l'automatiser totalement ou du moins partiellement.

Devenez des développeurs et apprenez les langages comme Python et Ruby. Beaucoup d'outils utilisés sont développés dans ces langages et cela vous facilitera grandement la création.

Si vous en avez la possibilité, mettez en place une solution pour donner accès aux logs applicatifs aux développeurs. Cela permettra d'accélérer la résolution des problèmes.

Diffusez les bonnes pratiques en en parlant à vos collègues, aux autres équipes et avec tous les Ops avec qui vous êtes en relation. À force de les diffuser, ces bonnes pratiques finiront par porter leurs fruits et faire changer petit à petit les mentalités.

> Quelle relation avec le Cloud ?

DevOps et Cloud vont de pair. Beaucoup de projets Cloud ont réussi comme ça. En effet il nous paraît très difficile de gérer un grand nombre de serveurs créés et détruits dynamiquement et à la demande, sans utiliser les pratiques et outils DevOps.

Un des avantages de la mise en place de DevOps c'est qu'il réduit considérablement les problèmes de réversibilité. Il devient possible de changer facilement de fournisseur Cloud ou de rapatrier votre application en interne sur votre infrastructure.

Certains outils comme SaltStack permettent également de provisionner des machines virtuelles.

> Quelques erreurs à éviter

Voici quelques erreurs qu'il est préférable d'éviter lorsqu'on commence à mettre en place les pratiques DevOps dans son entreprise.

Se concentrer uniquement sur les outils

Comme nous l'avons vu, DevOps, ce ne sont pas que des outils. Il y a une part importante de communication et de partage entre les équipes. Mettre en place uniquement des outils pour pallier à un manque de communication ne résoudra pas le problème.

Réinventer la roue

Vous avez certainement des processus et des bonnes pratiques qui existent déjà dans votre entreprise. Plutôt que de tout réinventer, commencez par essayer de les adapter à vos besoins.

C'est la même chose pour les outils. Regardez les outils et modules existants et développez par la communauté. Il y en a forcément un qui correspond à vos besoins.

Évitez un changement bigbang

Vouloir tout changer d'un coup est le meilleur moyen pour ne pas y arriver. Il faut mieux faire de petits changements réguliers et constater les améliorations.

Mettre un intitulé de poste DevOps

DevOps n'est pas un intitulé de poste. C'est une culture qui doit se répandre dans l'entreprise. Sans cette culture il ne peut y avoir de poste DevOps.

> Avantages et inconvénients de DevOps

Voici quelques-uns des avantages que vous

allez retirer en mettant en place les pratiques et outils DevOps dans votre entreprise. Vous allez pouvoir améliorer le Time to Market de vos applications tout en ayant une meilleure maîtrise de vos coûts. L'automatisation permet de réduire les erreurs et ainsi de diminuer les risques.

Le partage et une meilleure communication entre vos équipes feront réduire le stress lors des mises en production et feront augmenter leur productivité.

Cependant tout n'est pas tout rose et quelques inconvénients existent. Le principal étant le changement d'organisation que la culture DevOps implique. Ce sera le frein majeur que vous rencontrerez dans les grandes entreprises.

Les pratiques et outils DevOps étant relativement récents, ils ne sont pas encore enseignés à l'école et il n'existe pas encore beaucoup de profils ayant ces compétences.

> Conclusion

Dans le fil de cet article nous avons vu les pratiques et outils liés à la culture DevOps et comment les utiliser pour améliorer le Time to Market de vos applications. Au-delà d'être un terme à la mode, DevOps apporte de réels bénéfices pour l'entreprise. Même si cela prend du temps, DevOps va s'imposer dans nos entreprises comme les méthodes Agiles ont réussi à le faire.

Mais pour cela il faut diffuser cette culture, et personne d'autre que vous ne pourra mieux le faire.

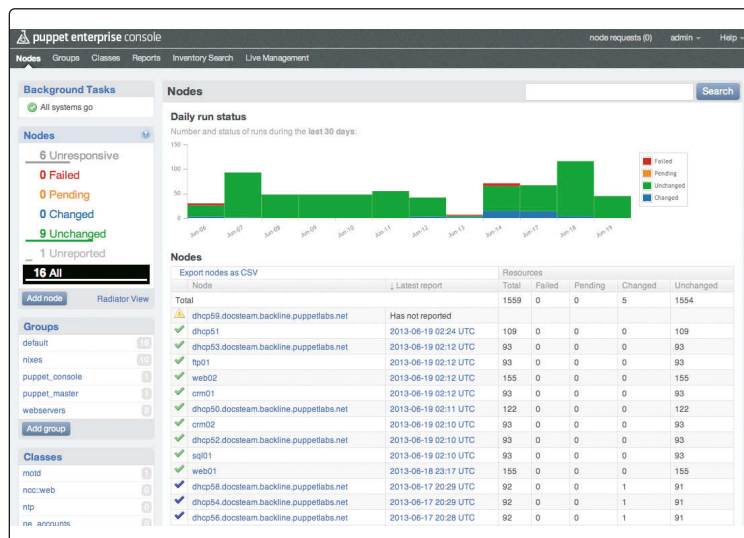
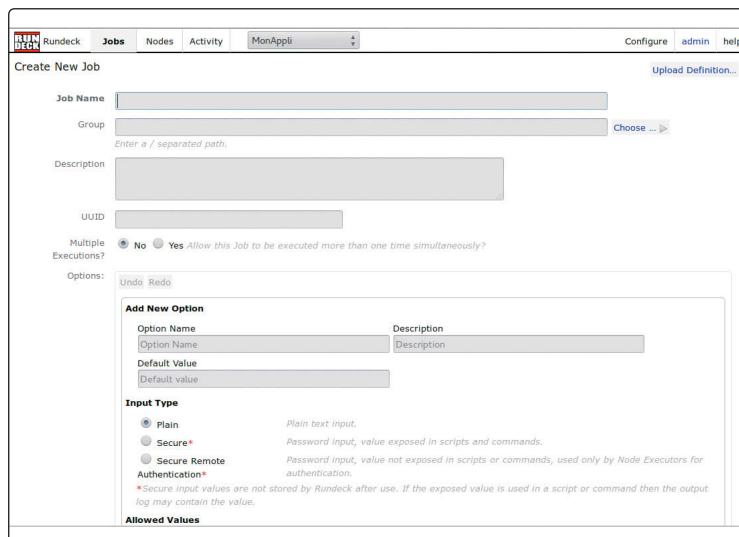
Yann Vigara

Directeur technique Ippon Hosting /

www.ippn-hosting.fr

Twitter : @YannVigara

Blog : blog.ippn.fr

The screenshot shows the 'Create New Job' form in the RunDeck interface. It includes fields for 'Job Name', 'Group', 'Description', and 'UUID'. There are also checkboxes for 'Multiple Executions?' and 'Options'. At the bottom, there's a section for 'Add New Option' with fields for 'Option Name', 'Description', 'Default Value', and 'Input Type' (Plain, Secure, Secure Remote). The form is designed for creating and managing jobs in the RunDeck system.



De l'intégration continue à la livraison continue

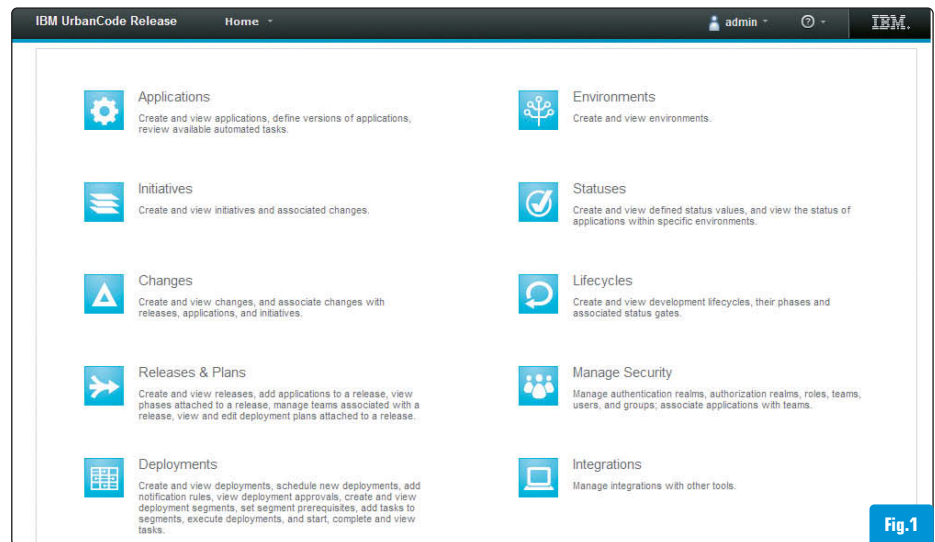
L'intégration continue est devenue une pratique systématique des équipes de développement. Elle permet d'automatiser les phases de compilation, assemblage, tests unitaires et tests d'intégration d'une application. L'intégration continue permet aussi d'orchestrer l'enchaînement automatique de ces phases, avec les itérations successives de codage, afin de délivrer les changements et les améliorations en continu.

Pour autant, à la fin du cycle d'intégration continue, l'application n'est pas encore en production. Avant d'y arriver, l'application transite successivement à travers un nombre variable d'environnements. Ceux-ci sont souvent tous différents afin de subir des tests et vérifications plus poussés, comme par exemple les recettes utilisateurs, les tests de montée en charge ou encore les vérifications de conformité. Durant ces phases, il est encore assez rare de trouver des processus continus tels qu'on les connaît dans le développement. Ceci est principalement dû au nombre important d'activités et à la diversité des tâches à exécuter, orchestrer et gouverner. Prédire et planifier ces tâches relève du tour de force pour beaucoup d'entreprises.

En effet, ces phases sont à la jonction de plusieurs départements et rôles, qui n'ont pas les mêmes priorités au jour le jour, et de fait ne sont pas alignés en terme de pratiques et processus.

Une solution réside dans la mise en place d'un processus de livraison continue, capable de réunir les différentes entités de développement, test et production. L'objectif d'un processus de livraison continue est de fluidifier l'ensemble des transitions d'une application dans la chaîne de livraison, à travers une orchestration planifiée, une visibilité centralisée et une traçabilité de bout-en-bout. Grâce à la livraison continue, les applications se retrouvent sur un 'tapis roulant' continu, permettant de traiter plus de livraisons, dans de bien meilleurs délais, comparé à une approche en 'trains' successifs.

IBM UrbanCode Release a été conçu pour fournir et gérer la livraison continue. Avec des fonctions centralisées d'orchestration, de traçabilité, de gestion de chaîne de livraison, cette solution permet de planifier et maîtriser les livraisons de chaque release qui



Page d'accueil de l'outil

peut être composée d'une ou plusieurs applications. Il permet aussi de définir les cycles de vie des processus de releases, gérer l'affectation des environnements, la progression des applications dans la chaîne...

Une autre constatation courante concernant les phases de livraison réside dans la complexité des opérations de déploiement à effectuer. Chaque phase de test ou de production est précédée de la mise en place ou mise à jour des infrastructures, et du déploiement des différentes applications en tout ou partie. Ces opérations sont très laborieuses, souvent effectuées manuellement, peu reproductibles et peu visibles. Elles sont souvent responsables d'une grande partie des goulots d'étranglement, de latences très longues, de frustrations majeures et de coûts disproportionnés.

Pour être efficace en plus d'être maîtrisée, la livraison continue doit donc s'accompagner de l'automatisation d'un maximum de tâches de déploiement. Mettre en place ou à jour un environnement complet pour effectuer une campagne de tests, ou vérifier la

conformité d'un système, peut et doit être automatisé. Cela commence par l'automatisation du déploiement d'une application sur un environnement, et peut aller jusqu'à automatiser le provisionnement des infrastructures ou la mise en place de robots ou bancs de tests.

IBM UrbanCode Deploy assure cette fonction d'automatisation des activités de déploiements applicatifs.

> Un survol de l'environnement UrbanCode Release

L'accès contrôlé aux fonctions se fait via un navigateur web, l'authentification se faisant ou pas en connexion avec un annuaire LDAP [Fig.1]. L'environnement propose une organisation en 3 niveaux avec une notion de plan de release, de plan de déploiement et de plan d'environnement.

Le plan de release (Release Plan) identifie quelles applications sont incluses dans une release. Des points de contrôle (Gates) et des cycles d'approbation sont définis pour assurer la qualité de ce qui est transféré

entre les différents environnements [Fig.2]. Le plan de déploiement (Deployment Plan) trace les changements au niveau des applications et des infrastructures. Il définit comment les changements sont introduits dans un environnement, qu'ils soient automatisés ou manuels. Il permet de gérer le déploiement des applications en les alignant avec les changements d'infrastructure inhérents à une release. Un plan de déploiement est constitué de segments qui sont des regroupements de tâches pouvant être liées les unes aux autres en cas de pré-requis. Par défaut, une entrée est proposée pour le segment des tâches de pré-déploiement et pour le segment des tâches de déploiement [Fig.3]. Le plan d'environnement permet de gérer les environnements associés à une release. Ceux-ci sont des environnements logiques auxquels sont associés des environnements physiques sur lesquels les applications sont déployées. L'environnement d'une release définit le champ d'interaction entre applications. Il inclut les applications qui ne font pas partie de la release mais qui sont nécessaires pour les tests.

> Un survol d'UrbanCode Deploy

Afin d'automatiser les opérations de déploiement qui peuvent l'être, la partie Deploy propose un ensemble d'outils parmi lesquels on trouve la notion de composant.

La définition d'un composant se fait principalement en précisant une liste d'artefacts qui doivent toujours être déployés ensemble, ainsi que le processus de déploiement de ces artefacts. La définition du processus de déploiement est effectuée grâce à un éditeur graphique qui permet de définir les différentes actions à exécuter pour le déploiement du composant et leur enchaînement [Fig.4].

Les actions sont construites en puisant dans la bibliothèque de fonctions prédéfinies qui est fournie en standard. Afin de favoriser une standardisation et une réutilisation maximale, cette bibliothèque peut être enrichie, grâce à l'architecture en plug-ins de la solution.

Une application est définie principalement en précisant une liste de composants, ainsi que l'enchaînement du déploiement de ceux-ci sous la forme d'un processus de niveau application [Fig.5].

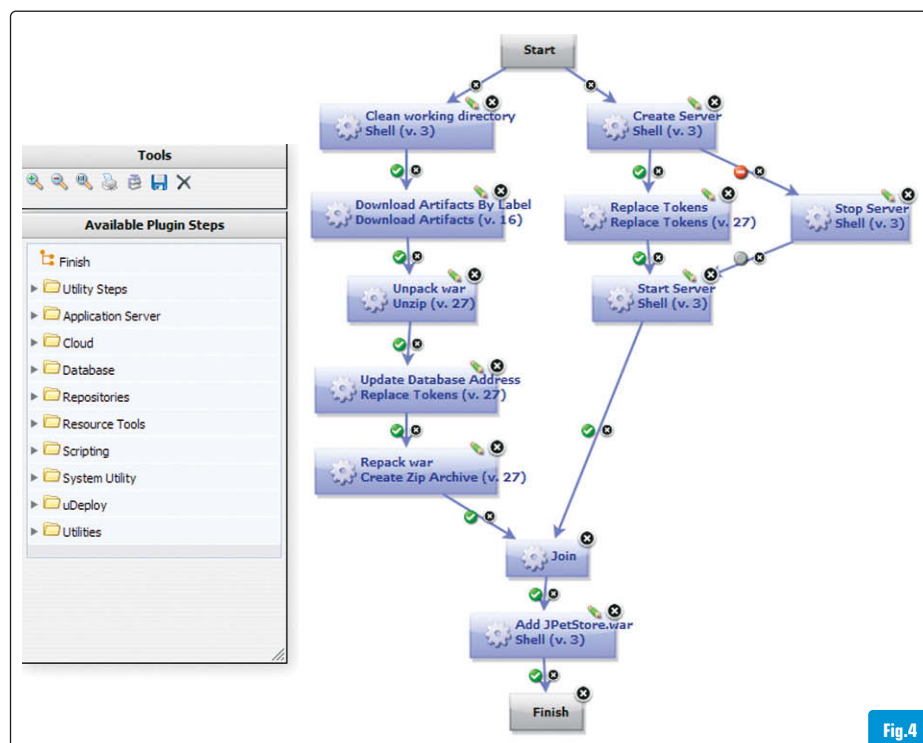
Des possibilités de paramétrage au niveau des composants, des applications, et des environnements permettent une bonne réutilisation des processus automatiques de déploiement. La solution permet de définir des processus de retour arrière indispen-

sables en cas de déploiement dans des environnements de production.

Le développeur dispose du concept d'image instantanée (Snapshot) afin de figer une version d'application et de la faire transiter d'un environnement à un autre. Cette image instantanée comprend la version du processus de déploiement associé à l'application, les versions des composants et les versions des processus de déploiements associés aux composants, ainsi que les versions des artefacts des différents composants.

L'outil versionne l'ensemble des objets manipulés, applications, composants, processus, snapshots. Les artefacts déployés sont versionnés et stockés dans le référentiel interne ou extraits d'un référentiel externe.

L'exécution d'un processus automatisé de



Exemple de processus de déploiement d'un composant

IBM UrbanCode Release

Deployments

admin

IBM

Home

Scheduled Deployments

Upcoming

All

Recurring

All Deployments

Add New

Execution Date

Release

Environment

Gate Status

Plan Status

Creation Source

Actions

8/1/13, 3:39 PM

March Release

PROD-1

Passed

In Progress

Manual

Application

Version

Statuses

Gate

March Release

Phase1

Passed

Save

Cancel

2/7/14, 3:52 PM

March Release

DEV-1

Passed

Not Started

Manual

Edit

2/28/14, 3:52 PM

March Release

QA-1

Passed

Not Started

Manual

Edit

3/28/14, 3:53 PM

March Release

PROD-1

Passed

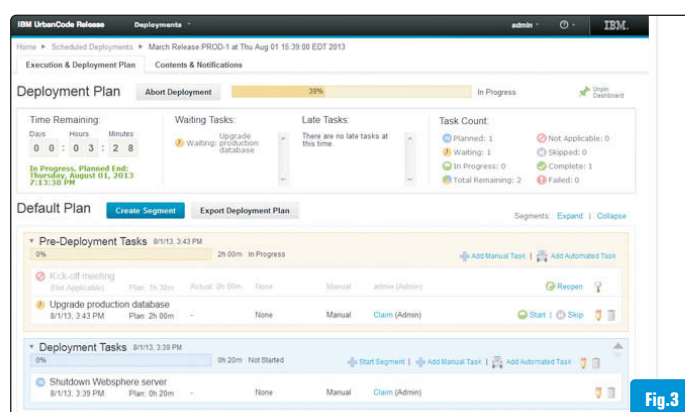
Not Started

Manual

Edit

Fig.2

Plan de release



Task	Planned	Actual	Status	Owner
Kick-off meeting (Not Applicable)	8/1/13, 3:43 PM	8/1/13, 3:43 PM	In Progress	admin (Admin)
Upgrade production database	8/1/13, 3:43 PM	8/1/13, 3:43 PM	Not Started	Claim (Admin)
Shutdown Websphere server	8/1/13, 3:39 PM	8/1/13, 3:39 PM	Not Started	Claim (Admin)

Plan de déploiement



déploiement est tracée de manière très fine avec des logs détaillés (détail de toutes les commandes exécutées, sorties standard, sorties d'erreur) qui permettent un débogage aisé en phase de mise au point et un suivi détaillé par la suite [Fig.6].

> Conclusion

L'implémentation d'un processus de livraison continue apporte des bénéfices tangibles et mesurables à toutes les entités d'une entreprise. Les départements IT de nos clients constatent des améliorations notables de leur productivité, une diminution de leurs coûts et une plus grande maîtrise des cycles de vie de leurs applications. Les entités métiers bénéficient de temps de mise sur le marché beaucoup plus rapides, accompagné de plus de prédictibilité et visibilité sur les cycles de release. Ils ont ainsi de bien meilleurs atouts pour planifier et délivrer leurs innovations sur le marché.

La livraison continue est une étape nécessaire pour les entreprises engagées dans une transformation orientée DevOps, et forme le socle pour aller plus loin vers la vérification continue et la prise en compte des retours utilisateurs en continu.

Samira Bataouche,
Ingénieur Consultant, IBM Rational.

Vincent Chartier
Architecte, IBM Rational.

PRÉCISIONS SUR LA GAMME

Avec IBM UrbanCode Release et IBM UrbanCode Deploy, IBM Rational complète sa plateforme Jazz (jazz.net) de gestion du cycle de développement des applications.

IBM UrbanCode Release V6.0 et IBM UrbanCode Deploy V6.0 sont les dernières versions commercialisées par la société UrbanCode avant son acquisition par IBM en avril 2013.

Pour en savoir plus sur l'offre logicielle IBM, des séminaires gratuits sont organisés régulièrement. Toutes les dates et descriptifs des contenus sont disponibles sur <http://www-05.ibm.com/fr/events/tec/>

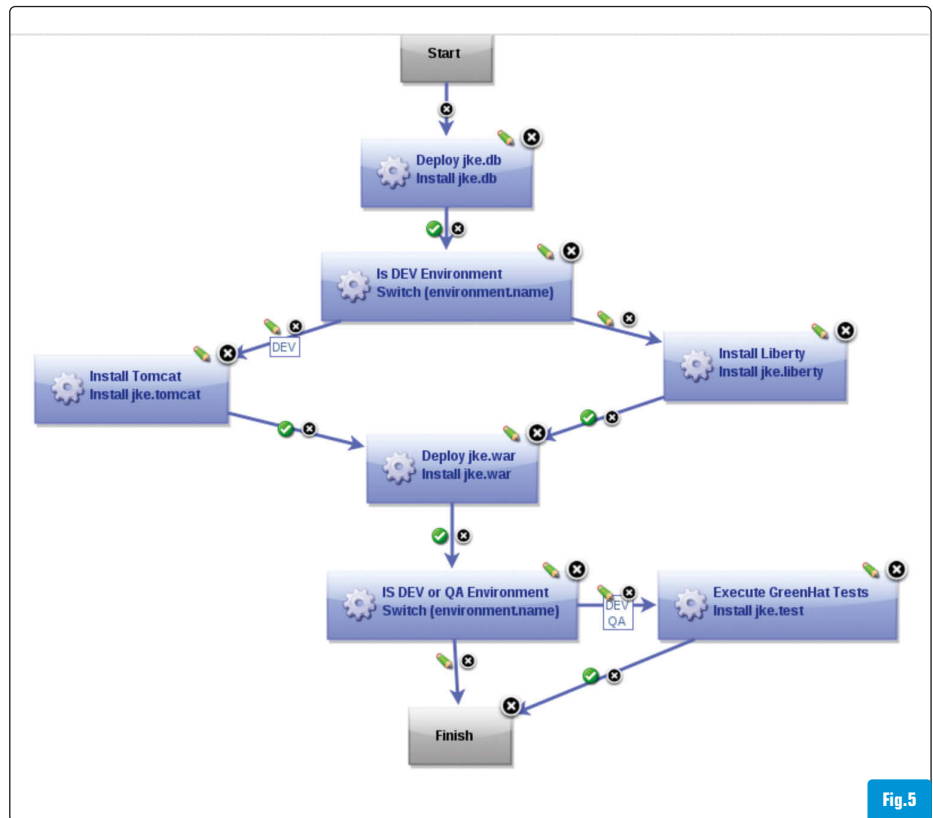


Fig.5

Exemple de processus de déploiement d'une application

IBM UrbanCode Deploy

Dashboard

Components

Applications

Configuration

Processes

Resources

Home > Applications > JKE > Process Request

Application Process Request: JKE

Process

Environment

Only Changed Versions

Date Requested

Requested By

Scheduled For

deploy (Version 3)

QA

false

09/12/2013, 06:44 PM

admin

09/12/2013, 06:44 PM

View Deployment Request

Log

Properties

Manifest

Expand All Collapse All Sort By: Graph Order Start Time

Step	Progress	Start
Deploy jke.db	1 / 1	6:44:16 PM
Switch: Is DEV Environment		6:44:38 PM
Install Tomcat		
Install Liberty	1 / 1	6:44:38 PM
Deploy jke.war	1 / 1	6:46:16 PM
Switch: Is DEV		6:46:51 PM
Validate JKE Deployment		
Total Execution	3 / 3	6:44:16 PM

Fig.6

Suivi du déploiement des différents composants d'une application



Définition et outillage du DevOps

DevOps est un sujet très large impliquant de nombreux outils et méthodes. Nous avons posé quelques questions à Sylvain Cailliau, directeur technique à SERENA France, éditeur ayant une large gamme d'outils adaptés à l'approche DevOps.

Programmez : quelle définition donnez-vous du DevOps ?

Au-delà des notions de « rapidité », la définition que SERENA donne du DEVOPS est : **La mise en œuvre d'un ensemble de bonnes pratiques et d'outils pour éviter les surprises (autant celles que les développeurs peuvent réserver aux exploitants, que celles que les exploitants peuvent réserver aux développeurs), et en finir avec les erreurs qu'on ne découvre qu'après la mise en production.**

Il en résulte effectivement que le passage d'une plate-forme à l'autre dans le chemin qui mène du développement jusqu'à la production (typiquement les environnements d'Intégration, de Recette, d'Homologation et de Pré-Production), est « fluidifié » par la mise en place de ces bonnes pratiques et de ces outils; la « bande passante » globale en est améliorée car il y a en conséquence plus de version (releases) plus rapidement. Cet effet (plus de Releases plus rapidement) peut être recherché au sein d'une entreprise comme principal bénéfice d'une démarche DEVOPS. Toutefois il ne sera pas obtenu si on cherche à

atteindre cet objectif sans prendre en compte les raisons fondamentales de la démarche (volonté d'en finir avec les erreurs en production par accroissement de la coopération entre les équipes de DEV et OPS). Il s'en suit que l'Automatisation des déploiements ne suffit pas ; tous les aspects communication sont fondamentaux. C'est bien ce que traduit le célèbre acronyme CAMS proposé par Damon Edwards et John Willis pour caractériser la démarche (Culture, Automation, Measure, Sharing). Nous pensons que la valeur de ces aspects communication peut-être maximisée si elle est formalisée dans des processus que

l'entreprise outillera pour garantir qu'ils sont systématiquement appliqués et que la bonne information est transmise vers les différents acteurs au plus vite.

Programmez : si on se place du point de vue du développeur, qu'est-ce que le devops change ou modifie au quotidien, dans la manière de mener des projets, de gérer les développements ?

Pour le développeur, au quotidien, dès lors qu'une démarche DEVOPS a été mise en place, le point majeur qui va modifier son quotidien c'est : la nécessité de ne pas « sur-

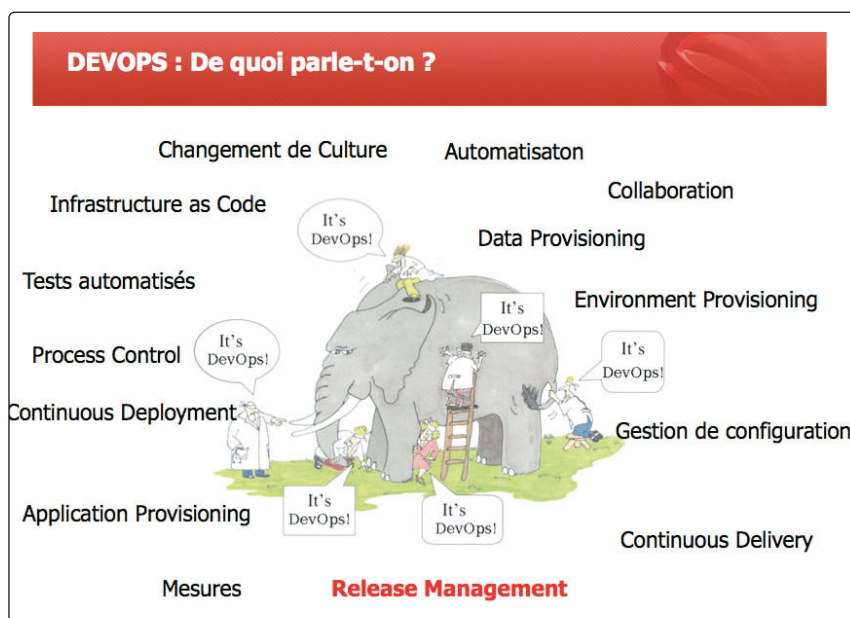
relèvent de leur responsabilité (mais pas sur cette plate-forme aval) ou pour demander aux exploitants un changement sur la plate-forme lorsque l'erreur ne leur est pas imputable.

On reproche souvent aux entreprises de toujours plus réduire les délais de développement, des tests, et de déployer toujours plus vite, n'est-ce pas un effet pervers du devops avec le risque de la précipitation ?

On a tous forcément en tête le cas de Knight Capital Group où un déploiement « trop rapide en production » provoque un « problème »

de 440 millions de dollars en 45 minutes. Mais la possibilité d'accélérer les mises en production n'est qu'un sous-produit de la démarche DEVOPS, son objectif principal est, au contraire, de prévenir les erreurs en production. De plus, la validation fonctionnelle est en dehors du domaine d'application de la démarche DEVOPS. Cependant en permettant une mise en service plus fréquente et plus rapide sur les plates-formes du chemin de la mise en pro-

duction DEVOPS permet de donner plus de temps pour réaliser ce qui est la raison d'être de ces plates-formes intermédiaires : les Tests. Une bonne pratique est de faire réaliser par les outils d'automatisation le plus de tests possible à l'issue de chaque déploiement. Les tests doivent couvrir tout le scope : tests d'assemblage (d'intégration), fonctionnels, de sécurité (intrusion), de robustesse, de performances, de reprise sur erreur ou sur désastre. Un bon nombre de ceux-ci peuvent être aujourd'hui automatisés. Avec DEVOPS, on ne devrait plus entendre de propos tels que : « bon comme



prendre » les équipes aval dans la démarche. Et si « surprise il y a », il ne faut pas que ce soit une « grosse surprise ». Il faut donc étendre les bonnes pratiques de l'Intégration Continue un cran plus loin, et déployer les applications sur la plate-forme suivante dans le cycle de vie. C'est-à-dire souvent et automatiquement en multipliant les tests automatiques à l'issue de ce déploiement. Il faut le faire en acceptant une contrainte de plus : les développeurs, s'ils ont accès à cette plate-forme, ce n'est qu'en mode « lecture » pour analyser les éventuelles erreurs et les corriger lorsqu'elles



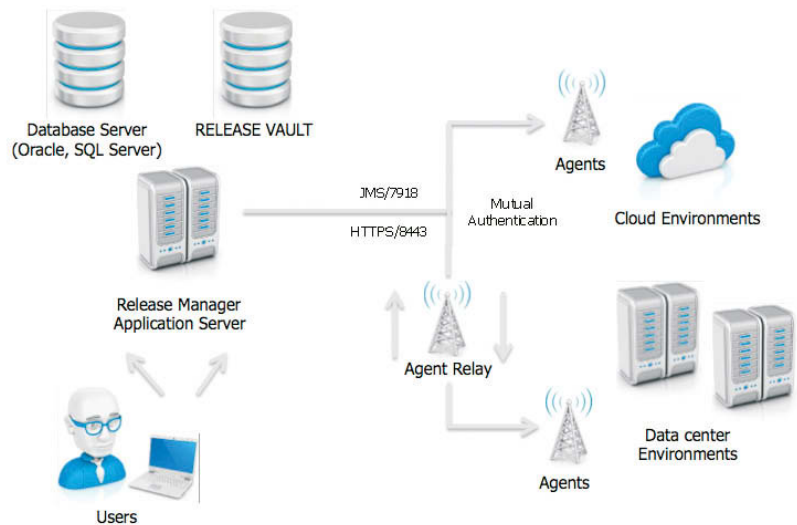
on a pris trois semaines de retard en intégration, on va limiter les tests fonctionnels ». Il faut aussi considérer un autre aspect dans la « culture » DEVOPS, l'approche Lean. Vous trouverez dans la littérature un autre acronyme CLAMS (ou CALMS) ou le L signifie Lean. C'est Jez Humble qui a introduit cet élément au sein de la démarche DEVOPS (et cet aspect Lean est très développé dans le livre « The Phoenix Project » de Gene Kim, l'ouvrage de vulgarisation du DEVOPS le plus célèbre). Cette approche applique au DEVOPS le principe du « No Waste » et du « Zero Stock » dans le cycle de vie applicatif. C'est en appliquant ces principes que l'on tire le meilleur parti d'une approche DEVOPS et que l'on obtient une accélération globale du processus. Et pas simplement parce qu'on a remplacé des activités manuelles par des automates.

Quels sont les outils de base pour mettre rapidement en place une approche devops ?

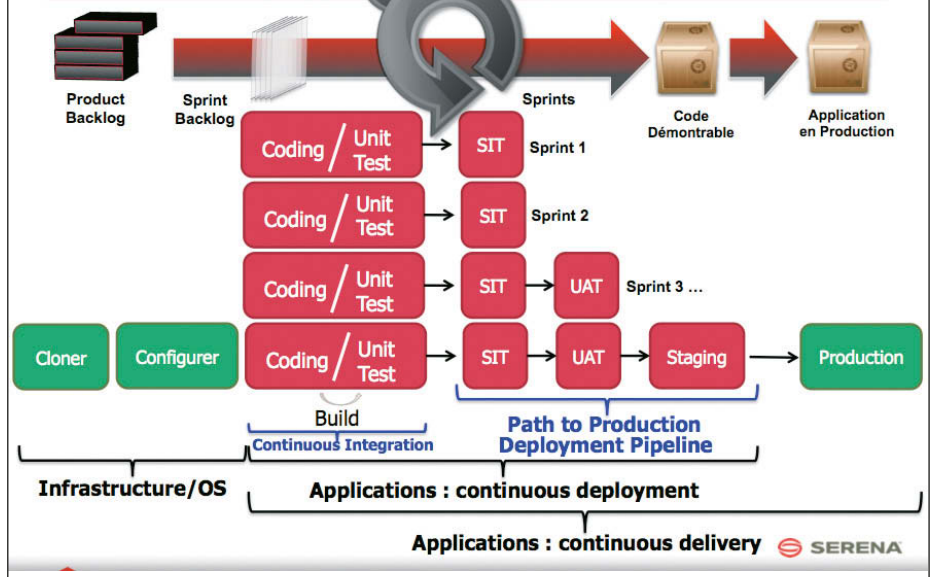
En terme d'outillage, il faut considérer les besoins d'une démarche DEVOPS comme relevant du provisionning et de la communication. Le provisionning s'opère à trois niveaux : provisionning Applicatif ; Provisionning de Données ; Provisionning d'Infrastructure. Ils sont ici classés par ordre de « fréquence » : on a plus souvent, dans la démarche DEVOPS, besoin de déployer une application, que de rafraîchir les données de tests (qui sont néanmoins fondamentales pour l'efficacité globale de la démarche), ou/et que de (re)construire une plate-forme. Les outils dont on va avoir besoin sont donc :

- Les outils d'automatisation du déploiement applicatif. Plusieurs offres sont disponibles sur le marché. SERENA pense qu'il est très important qu'une solution d'automatisation présente un méta-modèle qui permette aux développeurs de l'utiliser en y accédant via une « vue » familière. C'est-à-dire structurée par applications. Il faut aussi que ce méta-modèle permette aux exploitants d'aborder le déploiement selon une vue qui leur est plus adaptée : par environnements.
- Les outils d'automatisation de construction de jeux de tests. Ils permettent, à partir des données de production, de construire des jeux de tests adaptés. C'est-à-dire cross-plates-formes, anonymisés et sécurisés (les données sensibles sont protégées).
- Les outils de provisionning d'infrastructure. Ces outils sont particulièrement efficaces lorsque l'on peut traiter l'infrastructure

Architecture Serena Release Automation



Cycle de vie Provisioning



comme du « code » ; ce qui est le cas lorsque les environnements sont virtualisés ou dans un Cloud.

Par simple effet de répétition, choisir de débiter la démarche par le choix de l'Automatisation du Déploiement Applicatif (qui est réalisé plus fréquemment que les deux autres types de provisionning) s'avère le plus efficace.

Il faut, au plus tôt, mettre en place les outils de communication qui permettront d'automatiser tous les échanges entre les équipes, de manière systématiques (comme par exemple SERENA RELEASE CONTROL dans l'offre globale de Release Management

de SERENA). Ces outils implémenteront et garantiront l'exécution des processus qui auront été définis. Ils permettent d'alerter au plus vite les acteurs du processus par notification.

Enfin, trois pré-requis nous semblent essentiels : la mise en place en amont côté développement de l'Intégration Continue, la stricte ségrégation des artefacts invariants (un fichier .war par exemple) et des paramètres applicables par environnement (un fichier properties par exemple) dès le début des développements, l'utilisation d'un référentiel qui gère en versions les artefacts installables sur le Système d'Information.

DevOps, mise en pratique : Vagrant

Dans le dossier de ce mois-ci nous avons vu comment propager une culture DevOps dans votre entreprise et quelques-uns des outils disponibles. Nous allons voir au travers de plusieurs articles comment il est possible d'utiliser ces outils pour mettre en place du Continuous Delivery.

Votre mission si vous l'acceptez : déployer jusqu'en production l'application web java « mon-appli.war » que vous avez développée en ayant un minimum d'interventions humaines.

Vos armes d'automatisation massive : Git, Maven (3.0.x), Jenkins (1.535), Nexus (2.6.x), Vagrant (1.3.x), Puppet (3.2 minimum), Rundeck (1.6.x).

Votre infrastructure cible 4 serveurs Ubuntu Precise : un frontal Apache, deux serveurs Tomcat, un serveur PostgreSQL.

Nous allons partir du principe que vous disposez et avez déjà correctement configuré : un serveur Jenkins, un serveur Nexus, un serveur Rundeck, un serveur Puppet master, un repository central Git, un IDE, Git, Maven, VirtualBox et Vagrant installés sur les postes des développeurs et de l'exploitation.

Nous allons mettre en place le workflow de travail suivant : [Fig.1].

Les Dev et les Ops devront gérer le code source de leur développement et configuration dans un repository Git. Toutes les modifications du code source ou des scripts devront d'abord passer des tests dans plusieurs environnements (test, intégration, recette) avant d'être déployées en production.

Dans ce premier article nous allons voir comment utiliser Vagrant pour permettre de créer une infrastructure virtuelle identique à celle de la production.

> Définition de l'infrastructure virtuelle

Une des premières étapes pour pouvoir faire du Continuous Delivery consiste à tester le plus rapidement possible son application dans un environnement proche de celui de la production. Nous allons utiliser Vagrant pour créer cette infrastructure sur le poste du développeur.

> Introduction à Vagrant

Vagrant permet à partir d'un fichier descriptif de créer une infrastructure composée de machines virtuelles [Fig.2].

La description de l'infrastructure se fait via le fichier VagrantFile. Ce fichier permet de définir les images virtuelles de base ou box à utiliser et où les télécharger, la configuration réseau, les ports qui seront exposés en local. Pour démarrer rapidement un nouveau projet, il suffit d'utiliser la commande suivante :

```
$ vagrant init precise32 http://files.vagrantup.com/precise32.box
```

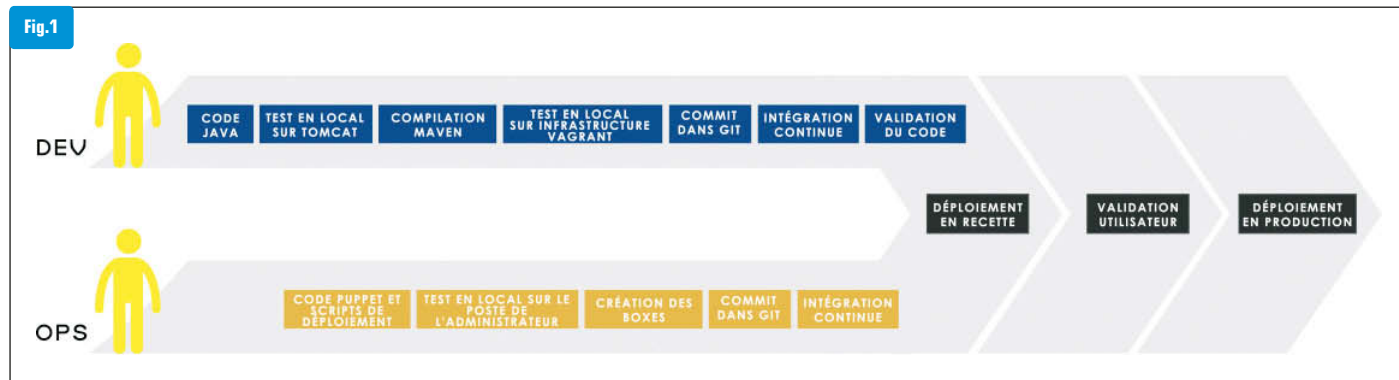
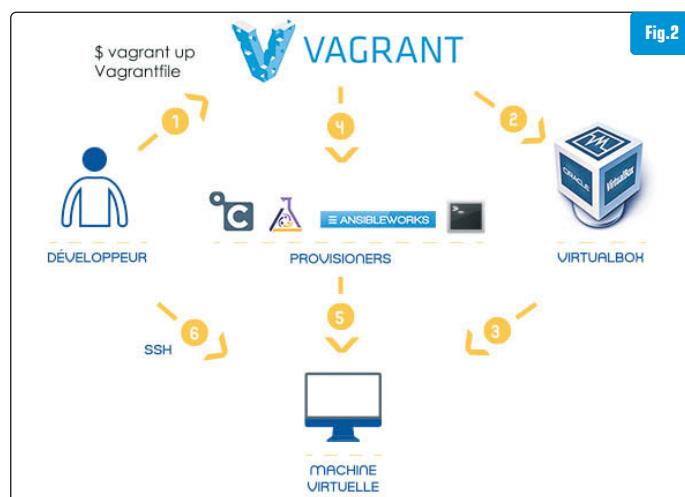
Cette commande va générer dans le répertoire courant un Vagrantfile minimal permettant de lancer une VM avec Ubuntu Precise 32bit. Le fichier ainsi généré aura la forme suivante :

```
Vagrant.configure(2) do |config|
  # nom de la box à utiliser
  config.vm.box = «precise32»
  # url pour télécharger la box ci-dessus
  config.vm.box_url = «http://files.vagrantup.com/precise32.box»
end
```

Le lancement de la VM se fait via la commande : \$ `vagrant up`. À ce moment-là, Vagrant va télécharger la box Ubuntu Precise 32 et la stocker dans votre « repository » de boxes disponibles si elle n'existe pas déjà (par défaut dans <home>/<vagrant>/<boxes>). Il va ensuite importer dans cette box VirtualBox, la configurer et démarrer la VM.

Pour voir l'état de la VM vous pouvez utiliser la commande : \$ `vagrant status`

Une fois la VM démarrée on peut s'y connecter à l'aide de la commande : \$ `vagrant ssh`





Vagrant se connecte à l'aide d'une clef SSH avec l'utilisateur vagrant. Cet utilisateur est présent sur les boxes mises à disposition par Vagrant. Il peut exécuter des commandes en root avec la commande `sudo` sans avoir à saisir de mot de passe.

Pour arrêter la VM rien de plus simple : `$ vagrant halt`

Et enfin pour supprimer la VM dans VirtualBox : `$ vagrant destroy`

Dans ce premier exemple, nous avons utilisé Vagrant pour démarrer une unique box. Vagrant permet de définir des infrastructures beaucoup plus complexes et composées de plusieurs machines virtuelles. Nous allons donc voir maintenant comment créer l'infrastructure qui sera lancée sur le poste des développeurs. Vagrant fournit des mécanismes permettant d'installer automatiquement des logiciels sur les VM pendant leur démarrage. Pour accélérer l'installation et le démarrage de l'infrastructure sur les postes des développeurs, nous allons leur fournir des box pré-configurées.

> Infrastructure complexe

Pour cela, nous allons créer deux infrastructures différentes. La première va permettre de tester la configuration Puppet et de créer les boxes qui seront utilisées par la seconde. Voici le fichier Vagrantfile : voir le code complet sur programmez.com

Ce fichier va configurer les 4 VM de l'infrastructure cible ainsi qu'une VM pour le puppetmaster. Nous utilisons dans ce fichier plusieurs fonctionnalités de Vagrant.

Configuration réseau

Tout d'abord toutes les VM devront être connectées sur un réseau privé. Elles devront pouvoir se voir et communiquer entre elles.

Vagrant offre plusieurs mécanismes de gestion du réseau. Ils sont paramétrables avec le paramètre de configuration « network ». Une VM peut définir plusieurs fois le paramètre network.

Dans notre cas nous utilisons pour toutes les VM un réseau de type « private_network » et nous définissons pour chacune d'entre-elles son adresse IP privée. Vagrant permet également d'exposer sur la machine hôte des ports configurés sur la VM. Pour cela il faut déclarer un « network » de type « forwarded_port ». Il faut ensuite définir le port sur la VM avec le paramètre « guest » et le port qui sera configuré en local avec « host ». Vagrant peut régler les conflits de ports qui pourraient survenir au moment du mapping.

Provisionnement de la configuration

La box que nous utilisons est une box générique installée avec un minimum de services. Comme nous voulons provisionner la configuration des serveurs à l'aide de Puppet, il est nécessaire de l'installer. Cette opération est réalisée à l'aide d'un script shell passé via le paramètre « provision ». Ce script sera exécuté uniquement lors du premier lancement de la machine virtuelle après son importation dans Virtualbox. Nous utilisons le même mécanisme pour installer et faire la configuration initiale du puppetmaster.

Vagrant propose d'autres mécanismes de provisionning. Il permet par exemple d'appliquer des configurations Puppet ou Chef définies en local, de se connecter à un serveur Puppet, Chef ou Ansible.

Répertoires partagés

Vagrant propose une solution pour partager un répertoire de la machine hôte et le rendre accessible par la machine virtuelle. Cela permet par exemple de continuer de travailler sur un projet avec son IDE pré-

féré tout en faisant exécuter le code sur la machine virtuelle. Nous allons utiliser cette méthode pour exporter la configuration Puppet vers le puppetmaster. Cette configuration sera gérée dans un repository Git. La configuration d'un répertoire partagé se fait à l'aide du paramètre « synced_folder ». Le premier paramètre correspond au répertoire local, le second au répertoire cible sur la machine virtuelle.

Démarrage et connexion aux VM

Comme dans le cas d'une infrastructure composée d'une seule VM, la commande `$ vagrant up` va créer et démarrer l'ensemble des VMs définies. Il arrive toutefois que l'on ne souhaite démarrer qu'une seule VM. Pour cela rien de plus simple, il suffit d'ajouter le nom de la machine en paramètre de la commande. Ce qui donne, pour ne lancer que la VM Front : `$ vagrant up front`.

En cas de doute sur le nom des machines à utiliser, la commande `# vagrant status` va afficher le nom et le statut de chaque VM.

Pour se connecter à une VM il suffit également d'ajouter son nom à la commande `$ vagrant ssh`.

Création des boxes pour les développeurs

L'infrastructure que nous venons de voir est avant tout à destination des Ops. Elle doit leur permettre de tester les configurations Puppet et de créer les boxes à destination des Dev.

La création d'une box s'effectue en exportant et en packageant une machine virtuelle en cours de fonctionnement. Vagrant s'occupe de faire cette opération. Il faut pour cela lui fournir le nom ou l'identifiant de la VM dans VirtualBox.

La commande `$ VBoxManage list vms` va retourner la liste des VM présentes dans VirtualBox. Vous pouvez ensuite exporter la box avec la commande suivante : `$ vagrant package --base <nom de la VM dans VirtualBox> --output <nom de la VM>.box`

Pour exporter la VM tomcat1 il faudra faire :

```
$ vagrant package --base devops_tomcat1_1382001140 --output tomcat1.box
```

Infrastructure de test

Maintenant que nous avons créé, configuré avec Puppet, et exporté nos boxes, nous allons pouvoir définir l'infrastructure qui sera utilisée par les développeurs pour effectuer les tests de l'application.

Pour faciliter le partage des boxes avec les développeurs, celles-ci devront être téléchargeables via un site web. Ce site peut être privé et accessible uniquement en interne.

La configuration sera très proche de celle utilisée précédemment. Nous utiliserons le Vagrantfile suivant pour notre infrastructure de tests : voir code complet sur programmez.com

Le Vagrantfile devra être à placer à la racine du projet. Lorsqu'ils compileront l'application avec Maven et qu'ils lanceront l'infrastructure, leur application sera automatiquement déployée. Cela est rendu possible grâce à l'utilisation d'un répertoire partagé entre les VM dédiées à Tomcat et l'hôte.

> Conclusion

Dans cet article nous avons vu comment créer et utiliser une infrastructure avec Vagrant. Dans le prochain article, nous verrons comment provisionner la configuration sur les VM avec Puppet.

Yann Vigara

Directeur technique Ippon Hosting / www.ippn-hosting.fr

Twitter : @YannVigara - Blog : blog.ippn.fr

A qui appartient "son" code source ?



Vous êtes-vous déjà posé la question ? Etes-vous propriétaire du code que vous écrivez ? Si vous récupérez des morceaux de code ici et là dans des exemples, sur le web, etc, quelles conséquences cela peut-il avoir pour vous ou l'entreprise qui va utiliser ces codes ?

Plusieurs affaires ont secoué le monde informatique et Internet : Oracle qui attaque Google sur la licence Java et le code source de ce dernier, Free et des problèmes liés à la licence GNU/GPL. Utiliser un code source d'origine indéterminée peut avoir des conséquences désastreuses.

Dura lex sed lex

La rédaction

La problématique

Par définition, tous les logiciels, tous les codes sources sont soumis à un statut juridique qui varie selon le pays : propriété intellectuelle, copyright, droit moral, droit patrimonial, brevet logiciel, etc. Nous n'avons que l'embaras du choix.

Bref, tout n'est pas permis. Vous ne pouvez pas réutiliser un code source dans un logiciel commercial / payant si ce morceau de code est soumis à une licence d'utilisation précise. Dans certains cas, vous pouvez librement récupérer, modifier et redistribuer un code. Il existe de très nombreuses licences open source et propriétaires, avec des permissions et des contraintes très différentes.

Déterminer l'origine d'un code récupéré sur Internet ou auprès d'un collègue n'est pas à prendre à la légère surtout pour des entreprises sensibles à la propriété intellectuelle. La moindre « faille » peut coûter très cher. Si un code à l'origine indéterminée est utilisé, mieux vaut parfois le redévelopper entièrement pour éviter toute ambiguïté.

Autre question qui revient souvent, « qu'est-ce que je fais quand il y a des codes soumis à différentes licences ? » Quelle licence prime sur l'autre ? Faut-il se référer à la licence la plus permissive ou la plus contraignante ?

Les outils de recensement et d'audit open source : pour quoi faire ?

Pour pouvoir recenser les licences utilisées dans les codes sources, un outil de recensement d'audit est indispensable quand un package applicatif représente plusieurs centaines ou milliers de fichiers.

Ces outils analysent l'ensemble des fichiers indiqués. Le moteur d'analyse utilise des bases spécifiques pour les licences et les codes référencés (base de connaissance incluant les forges). C'est en quelque sorte une empreinte avec les caractéristiques de la référence. Ces outils supportent les codes sources, les codes objets, les exécutables. L'objectif est de générer un rapport, le Bill of Material (BOM). Ce rapport recense tous les éléments utilisés dans le projet : l'application.

Pour faciliter le travail de ces outils et mieux identifier les codes et licences, un standard a été défini par la Linux Foundation, le SPDX. Même si l'initiative est assez récente, ce format se répand. Le fichier SPDX comprend de nombreux éléments (nom du package, nom du fichier, description, identifiant, licence déclarée, etc.). Il est au niveau du package et de chaque fichier.

> Quelques outils

Blackduck : référence et leader du marché ! Il est utilisé par les grandes entreprises pour scanner les projets et logiciels.

Code Janitor : initié par la fondation Linux. Analyse les sources et les licences des fichiers de vos projets.

Antepedia Tool Suit : outil payant mais gratuit pour les projets non commerciaux. Est constitué de deux composants : Repor-

ter (analyseur) et Notifier qui se connecte à un outil de gestion de code pour détecter les codes open source en temps réel. Site : <http://www.antelink.com>

Fossology : outil d'analyse statique, projet open source initié par HP. Reconnu par la communauté et qui évolue régulièrement. Site : <http://www.fossology.org>

> Quels usages ?

extraits du livre blanc « les outils de recensement et d'audit open source » (Smile)

Politique open source

Les bonnes pratiques de la *politique open source*, encore appelée *gouvernance open source*, mettent toujours en avant une première étape, qui est d'identifier l'existant en matière d'open source dans l'entreprise. Et effectivement, une majorité d'entreprises n'ont pas une vision complète des logiciels déployés au sein du système d'information. L'open source entre souvent par la petite porte, simplement parce qu'un administrateur système, un chef de projet, un développeur, a jugé qu'il pouvait gagner du temps et de l'argent, en intégrant tel composant qui répond à ses besoins. Pas de bon de commande, pas d'autorisation requise.

Or méconnaître son utilisation de l'open source est dommageable, pas uniquement sous l'angle juridique. Peut-être que parmi ces composants, certains seront obsolètes,

ou présenteront des failles, ou n'auront pas le niveau de support souhaité. Bref, identifier son patrimoine est toujours une bonne chose, le point de départ de sa politique open source. C'est aussi, comme on l'a dit, l'occasion de mesurer ce qui a été gagné déjà par ces composants, le plus souvent gratuits. Le recensement porte sur des référentiels de code déjà clairement identifiés.

Avoir une vision claire

Le besoin de voir clair dans son patrimoine de code est très réel. Le développement moderne repose largement sur un assemblage de composants. Un programme peut en compter des milliers. Il n'est pas aisé de garder précisément la trace de l'origine de chacun d'entre eux et de ses conditions de licences. Les programmeurs ne sont pas tous bien formés sur le sujet, et une partie du logiciel peut être sous-traitée à différents fournisseurs dont le niveau de maîtrise de la propriété et des licences peut être insuffisant.

Lors de fusion et acquisitions : auditer le parc logiciel

Ces questions surviennent souvent à l'occasion des acquisitions d'entreprises dont une grande partie du patrimoine est fait de logiciels. L'acquéreur se demande naturellement d'une part à qui appartient – de manière détaillée – ce logiciel, et d'autre part ce qu'il est autorisé à en faire. Cela peut concerner un éditeur de logiciel, bien sûr, mais aussi un fabricant de produits physiques incluant du logiciel embarqué, ou bien encore une startup du web. Quelle part du logiciel appartient à l'entreprise, et pour ce qui ne lui appartient pas, dispose-t-elle des droits d'utilisation requis ? L'acquéreur sera-t-il libre de faire un autre usage de ces logiciels ? L'acquéreur court-il le risque d'être attaqué par le détenteur des droits de l'un des composants logiciels ?

Sans oublier la prévention du risque juridique, les exigences lors de la commercialisation de logiciels, de services (pour lever tout doute sur la conformité).

Le problème du code snippet

Un simple copier-coller de quelques dizaines de lignes de code implique une œuvre dérivée, et induit donc l'application des obligations de certaines licences, dites copyleft. Détecter un copier-coller est bien plus complexe et bien plus incertain que détecter l'égalité stricte de composants logiciels entiers. Rechercher l'insertion d'un nombre quelconque de lignes coupées au sein de l'un des milliers - voire millions - de composants du référentiel est pratiquement impossible à traiter de manière exhaustive. Et plus encore si l'on imagine détecter la manœuvre d'un programmeur qui chercherait à cacher son méfait, et pourrait donc avoir modifié quelques commentaires, voire quelques noms de variables.

(extrait livre blanc « les outils de recensement et d'audit open source », Smile)

A qui appartient mon code source ?

Rarement le développeur se pose la question : à qui appartient le code que j'écris ? Il paraît si évident que le code écrit en entreprise appartient à l'entreprise et que le code écrit par moi m'appartient. Mais est-ce aussi évident que cela ?

Le développeur pioche du code source, des extraits ici et là. « C'est une bonne chose. Cela fait gagner du temps et de l'argent » précise Patrice Bertrand (Smile). Cela évite de réécrire un code récurrent ou de réinventer la roue à chaque projet. L'open source favorise cela ainsi que les bibliothèques de codes (comme les codes snippets). Mais cette réutilisation ne signifie pas que l'on peut tout faire. Il faut respecter certaines règles comme les licences et/ou la propriété intellectuelle.

Pareillement, il y a des règles à respecter sur la programmation dans le cadre du projet professionnel, en entreprise ou en SSII, et le travail communautaire ou personnel. « Quand on explique, les développeurs comprennent. » poursuit Patrice Bertrand. Mais les connaissances sur les problématiques légales et juridiques sont souvent très rudimentaires voire inexistantes. On ne demande pas au développeur de connaître la jurisprudence mais d'être sensibilisé à ces problèmes pour bien comprendre ce qu'il peut faire, ce qu'il ne peut pas faire et pour qu'il respecte certains principes. Il est dommage que les cursus et les formations n'incluent pas automatiquement un module sur ces sujets...

> Droit moral vs droit patrimonial : « la théorie »

Tout auteur, tout créateur possède de facto le droit moral. Il s'agit d'un des droits fondamentaux de la propriété intellectuelle en France. Il est attaché à sa personne. Il est perpétuel, inaliénable et imprescriptible. Ainsi, vous ne pouvez pas céder ou renoncer à ce droit. Il peut se faire reconnaître son droit à la paternité, en qualité d'auteur. Il possède aussi le droit au respect de son œuvre. Il peut s'opposer à tout usage, toute modification, suppression, ajout, pouvant modifier son œuvre originale. Enfin, il peut aussi retirer son œuvre... Cette définition est celle donnée par la SACD. Mais cela ne signifie pas que l'auteur possède le droit d'exploitation et d'en tirer de l'argent. Il garde sa qualité d'auteur du code, et peut

éventuellement exiger l'arrêt de l'utilisation d'un code, d'un logiciel dont il est l'auteur car son utilisateur le heurte.

Le droit patrimonial est, pour simplifier, l'exploitation du code, du logiciel. Ainsi, si le développeur travaille pour une SSII, un éditeur, une entreprise, il en est donc salarié et reçoit un salaire. L'exploitation sera alors assurée par l'entreprise qui va vendre, distribuer, maintenir. Mais chaque développeur garde un droit moral sur le code, le design qu'il réalise. Mais en pratique, ce droit est amoindri. « Il y a dévolution automatique (du logiciel) à l'employeur. L'entreprise et/ou le salarié peuvent prévoir le contraire. » recadre Maître Fabrice Perbost (avocat associé chez Kahn & Associés). Le développeur salarié cède à l'employeur les droits sur le code créé. Pour un stagiaire, la situation est différente car le stagiaire n'est pas un salarié. La convention de stage doit alors préciser ce « détail ».

L'article L113-9 du code de la propriété intellectuelle est très précis : « *Sauf dispositions statutaires ou stipulations contraires, les droits patrimoniaux sur les logiciels et leur documentation créés par un ou plusieurs employés dans l'exercice de leurs fonctions ou d'après les instructions de leur*

“ Êtes-vous sûr de connaître l'origine des codes et bibliothèques utilisés ? ”

Q&A

• Si je développe des projets personnels ou communautaires, mon employeur peut-il m'interdire de coder ?

C'est une question qui revient souvent : vous êtes développeur et vous codez chez vous le soir, le week-end, durant les vacances à des projets communautaires ou à des applis personnelles qui sont soumis à un App Store ou vendus : mon employeur peut-il me l'interdire ou limiter mes développements ?

Pour Pierre Ficheux d'OpenWide, si le projet personnel ou communautaire est très différent de votre travail professionnel, et qu'il n'y a pas de concurrence, oui vous pouvez. Mais comme le rappelle Maître Fabrice Perbost, en apporter une preuve matérielle est difficile. Et il existe toujours un risque à utiliser les connaissances acquises chez votre

employeur ou à utiliser un code créé lors d'une mission.

Le contrat de travail doit être le plus clair possible sur ce point. Vous pouvez aussi demander des clauses spécifiques à votre contrat pour éviter toute ambiguïté.

Nous vous conseillons de prévenir votre employeur de votre activité communautaire ou personnelle.

• Si je code durant le déjeuner pour un projet personnel dans les locaux de mon employeur, qu'est-ce que je risque ?

Sauf clauses spécifiques dans votre contrat ou dans le règlement intérieur, tout développement personnel dans les locaux de l'employeur peut être interdit, engendrer un blâme, etc. Le code écrit appartient à l'employeur dès lors qu'il est rattachable à l'activité de l'entreprise et à l'activité normale du

salarié. N'utilisez pas un ordinateur de l'entreprise pour le faire, et l'usage d'un matériel personnel n'est pas toujours autorisé. Nous préconisons une totale transparence de votre part et de la part de l'employeur. Les règles doivent être clairement établies et affichées. Toute ambiguïté n'est pas bonne.

• Quand je pars de l'entreprise, quel que soit le motif, je peux emporter les codes que j'ai développés ?

Non. Sauf clauses ou accords écrits, le code appartient à l'employeur. Vous ne pouvez pas les copier, les récupérer, les réutiliser.

• Si le code vient de l'étranger, je peux tout faire avec ?

Non ! Il est lui aussi protégé même si la notion de copyright est différente du droit intellectuel.

employeur sont dévolus à l'employeur qui est seul habilité à les exercer». « En matière de logiciel, le droit moral est bien plus limité pour ne pas freiner l'industrie logicielle et ne pas avoir un développeur omnipotent. En pratique, il est ainsi rare que les développeurs fassent valoir leur droit moral. » précise Maître Fabrice Perbost. Ainsi dans le cadre d'un développement sur mesure par une SSII, le code source est livré avec le projet terminé. Le client est le propriétaire du logiciel et du code. Attention : « il faut préciser que le code source est cédé » rappelle Maître Fabrice Perbost.

> Attention à la licence liée aux codes extérieurs

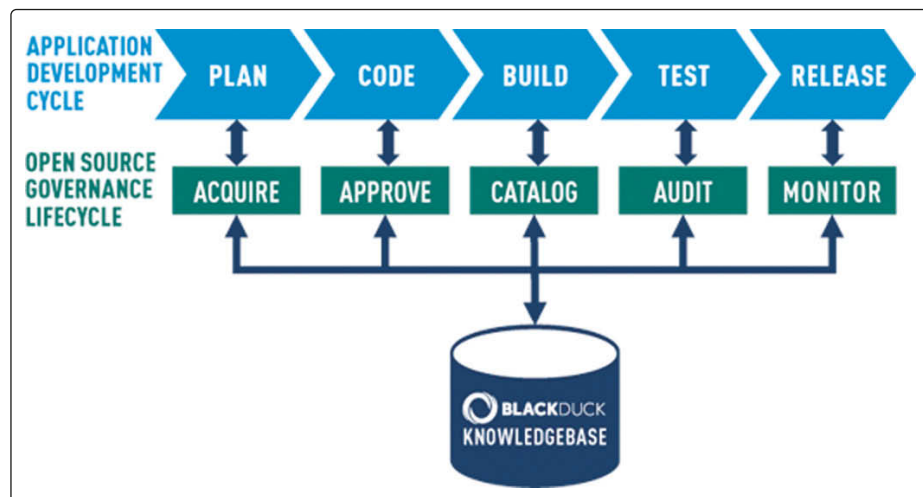
Le développeur ne sait pas forcément quelle licence s'applique à tel code, surtout quand les projets se composent de plusieurs centaines de fichiers différents, provenant de

différentes équipes. L'utilisation d'outils spécialisés permet de faciliter le travail d'identification. Ces outils aident à déterminer si une licence contaminante est présente ou non dans un source utilisé. Les problèmes potentiels existent quand l'application est vouée à être mise sur un App Store, ou tout autre canal de distribution. Les éditeurs sont très vigilants sur la légalité du code source. Le but n'est pas d'interdire l'usage de codes externes par les développeurs mais d'encadrer leur usage. Il faut pouvoir suivre et identifier l'origine de tout code externe. Et de décider s'il est possible de l'utiliser ou non. Le développeur ne pense pas toujours à

inclure les licences ou à renseigner dans le code les droits et les précisions légales utiles, même sur un code écrit en interne. Or, cela facilite la future traçabilité. De même, si un code externe arrive avec les informations légales, ne les supprimez pas. Vous devez les inclure dans votre projet. L'usage d'outils comme Blackduck revient parfois très cher. Les SSII n'utilisent pas toujours ces environnements laissant aux clients le soin de réaliser l'audit final. Par contre, il peut être précisé par contrat que tout usage de licences de type GPL doit être signalé aux clients.

François Tonic

aclocal.m4	[View][Info][Download][Tag]
GPL Licenses (Demo)	
AUTHORS	[View][Info][Download][Tag]
non-gpl (Demo)	
AUTHORS.pb	[View][Info][Download][Tag]
non-gpl (Demo)	
bootstrap	[View][Info][Download][Tag]
non-gpl (Demo)	
ChangeLog	[View][Info][Download][Tag]
GPL Licenses (Demo)	
compile	[View][Info][Download][Tag]
GPL Licenses (Demo)	
config.guess	[View][Info][Download][Tag]
GPL Licenses (Demo)	
config.h.in	[View][Info][Download][Tag]
non-gpl (Demo)	
config.sub	[View][Info][Download][Tag]
GPL Licenses (Demo)	
configure	[View][Info][Download][Tag]
GPL Licenses (Demo)	
configure.in	[View][Info][Download][Tag]
non-gpl (Demo)	
COPYING	[View][Info][Download][Tag]
GPL Licenses (Demo)	
depcomp	[View][Info][Download][Tag]
GPL Licenses (Demo)	
docs/	[Tag]
GPL Licenses (Demo), non-gpl (Demo)	
Doxyfile	[View][Info][Download][Tag]
non-gpl (Demo)	



OFFREZ-VOUS LE MONDE DU DÉVELOPPEMENT

Abonnez-vous jusqu'à **-50%**

et pour **1€** de **+** recevez un "livre numérique" des Éditions ENI

Coupon d'abonnement page 21 et sur www.programmez.com/abonnement.php



Logiciel = code source ? La réponse du juriste

Si on devait demander à un programmeur de réaliser un logiciel, ce programmeur se mettrait à... programmer, bien sûr ! Il écrirait, ce que l'on appelle, un code source, c'est-à-dire un ensemble d'instructions données à la machine dans un langage compréhensible par l'homme (C#, PHP, Python, etc). Peut-être notre programmeur ne ferait-il donc pas une réelle distinction entre le source et le logiciel qui le sous-tend. Et pourtant, les juristes – dont le langage est parfois aussi hermétique que celui des informaticiens – distinguent bien les deux.

Où est la différence ? La trouve-t-on dans la loi ? Chacun sait sûrement qu'un logiciel est protégé principalement par le droit d'auteur. Mais, si on parcourrait le Code de la propriété intellectuelle (www.legifrance.gouv.fr), on ne trouverait pas la réponse. La loi parle bien de logiciels et de bases de données, mais à aucun moment de code source. Et on pourrait feuilleter toute la législation que le résultat ne serait pas différent.

> La protection du code source

Pour comprendre la différence, on peut employer une comparaison avec une œuvre d'art, une sculpture en particulier : celui qui achète la sculpture pour la mettre dans son salon ne dispose pas forcément du moule qui lui permettrait d'en produire une seconde pour son jardin. En fait, le code source d'un logiciel est un peu ce que le moule est à la sculpture : une matrice, une machine à dupliquer, à répéter, à reproduire. Avec une différence toutefois, c'est que le moule d'une sculpture n'est plus modifiable, contrairement au source. Mais alors, sur quoi porte au juste la propriété intellectuelle ? Sur la sculpture ou sur le moule ? Sur les deux en fait. Le droit d'auteur protège le logiciel, quel que soit la forme sous laquelle il se présente. Le droit d'auteur protège à la fois le code source dans sa forme « littéraire », mais aussi dans sa forme exécutable. La protection du code source est acquise, même s'il n'est pas fini et même s'il ne fonctionne pas, parce que le droit d'auteur est totalement indifférent aux fonctionnalités du logiciel. En théorie, on peut donc agir en contrefaçon pour sanctionner la reproduction d'un logiciel complètement buggé ou, plus vraisemblable, inachevé. Là où les choses se compliquent, c'est qu'en pratique, l'utilisateur d'un logiciel ne dispose que d'une version exécutable du code source, le plus souvent compilée, c'est-à-dire illisible par l'humain. En règle générale, on ne donne pas le source à l'utilisateur final. C'est donc qu'au-delà de la question de la propriété intellectuelle sur le code, il y a surtout un enjeu en termes d'accès. Celui qui a accès

au code source emporte la maîtrise du logiciel. C'est pourquoi il est souvent crucial de savoir qui va accéder au code source.

> L'accès au code source en entreprise

L'accès, c'est d'abord la compréhension du code source, qui doit être codé dans les règles de l'art. Adieu les variables du type \$a12 ou \$t3... Si plusieurs programmeurs travaillent sur un même projet, il faut que l'accès au source soit facilité par des commentaires dans le code. En entreprise, ces questions sont souvent traitées dans une charte informatique rédigée par l'employeur. L'accès au source, c'est donc en premier lieu, l'accès des collègues à ce que je programme. En second lieu, c'est ensuite l'accès de l'employeur au source. Un programmeur salarié est certes un auteur, mais il est moins bien traité qu'un écrivain ou un compositeur. Le contrat de travail du programmeur emporte une cession automatique de ses droits sur ses logiciels, s'il a bien reçu une mission de développement. Mais la cession des droits emporte-t-elle le droit pour l'employeur d'accéder au code source ? La réponse est oui, bien entendu, mais c'est plutôt le droit du travail qui l'exige que celui de la propriété intellectuelle. Le contrat de travail est une location de la force de travail. Tous les produits issus du travail appartiennent à l'employeur, à plus forte raison quand le salarié agit sur ses instructions, avec les moyens de l'entreprise. Le plus souvent, le source est développé sur des ordinateurs mis à disposition du salarié par l'employeur, de sorte qu'ils ont un caractère professionnel. L'employeur est assurément en droit d'accéder librement au source développé par ses programmeurs salariés, dont il acquiert automatiquement les droits d'exploitation par ailleurs. Malheureusement, la loi n'a pas vraiment bien fait les choses. L'employeur n'est propriétaire que du logiciel stricto sensu. Pour tous autres éléments du logiciel, il est censé obtenir la cession des droits auprès de ses salariés ou freelance. Il est tout de même curieux que le législateur

n'ait pas étendu la présomption de cession des droits des salariés aux bases de données, ni d'ailleurs aux interfaces graphiques... La loi a échoué à énoncer des règles s'appliquant au logiciel dans sa globalité technique. Sauf exception, les juristes sont obligés de dépecer le logiciel, selon qu'il y a ou non présomption de cession. Franchement compliqué !...

> L'accès au code source par l'utilisateur

En troisième lieu, se pose la question de l'accès de l'utilisateur au source. Il est clair que la question ne se règle pas de la même manière selon diverses variables : selon que le logiciel est un progiciel ou un spécifique, selon que le logiciel est open source ou non, etc. La crainte de l'utilisateur sans accès au source, c'est de voir l'éditeur de son logiciel disparaître, sans pouvoir poursuivre l'utilisation normale du logiciel. C'est pour prévenir ce genre de situation que l'on propose souvent de conclure un contrat de séquestre avec l'utilisateur. Le principe en est très simple : en cas de liquidation judiciaire de l'éditeur, l'utilisateur peut réclamer l'accès au source auprès d'un tiers indépendant (l'APP, par exemple : <http://www.app.asso.fr/>). Parfois, on peut aussi prévoir un accès au source en cas de défaillance prolongée dans la maintenance. Ces questions sont étroitement liées à l'autorisation ou non donnée à l'utilisateur de recourir à la tierce maintenance pour exploiter le logiciel. La loi prévoit que l'utilisateur a le droit de corriger les erreurs d'un logiciel qu'il utilise, mais que l'auteur de ce logiciel peut toutefois le lui interdire par contrat. On l'aura compris, le source est une ressource dont l'accès doit être précisément encadré. Dans le silence de la loi, un seul outil pour cela : le contrat.



David Lefranc,
Avocat spécialisé en droit des nouvelles technologies, Cabinet LAROPAIN (ARRAS), Docteur en droit, Enseignant
Email : conseil@laropain.com
Web : <http://www.laropain.com>

Maîtriser l'asynchronisme de C# 5.0 ^{1^{re} partie}

Une des nouveautés majeures de C# 5.0 est sans aucun doute l'intégration de l'asynchronisme au niveau du langage. Avec cette offre, il devient difficile de produire des applications peu ou pas disponibles aux yeux de l'utilisateur. Cependant, l'expérience montre que cette grande nouveauté n'est pas toujours bien comprise et que des erreurs sont parfois difficiles à corriger lorsque le développeur n'a pas conscience des mécanismes sous-jacents. L'objectif de cet article est de répondre à quelques questions : pourquoi a-t-on besoin d'asynchronisme ? Comment définir une interface asynchrone simple et facile à maintenir ?

MOTIVATIONS

> Latences et satisfactions de l'utilisateur

Depuis de nombreuses années, la latence dans les programmes informatiques est un sujet grandissant. Dans un premier temps les applications graphiques, puis les applications Web, dernièrement le Cloud et enfin les tablettes et les smartphones, doivent offrir un niveau de disponibilité maximum afin que les utilisateurs ne souffrent pas de délais qui pénaliseraient leurs usages et leur satisfaction.

Sur le plan technique, la latence a deux origines : un usage intensif du CPU ou des entrées/sorties poussives (disque, réseaux...). Pour résoudre ce problème de latence, nous devrions appeler « en asynchrone » toutes les méthodes susceptibles d'être lentes.

Pour éviter toutes ambiguïtés voici comment nous pouvons définir un appel synchrone et un appel asynchrone :

- un appel synchrone : l'appelant ATTEND que la méthode se termine. Si l'appel est long, l'application reste figée et n'est plus disponible pour répondre à de nouvelles demandes.
- Un appel asynchrone : la méthode retourne immédiatement vers l'appelant et exécute un callback (ou une continuation) lorsque la méthode est terminée. L'appelant n'est pas bloqué, il reste disponible pour répondre à d'autres demandes.

> Asynchronisme dans notre quotidien

Avant de détailler comment fonctionne l'asynchronisme de C# 5.0, je vous propose une petite histoire tirée de la vie réelle. Vous êtes chez vous et vous venez de constater que votre connexion Internet ne fonctionne plus. Vous décidez d'appeler votre opérateur Internet afin de résoudre le problème. Vous composez le numéro du support de votre opérateur et vous obtenez ce message vocal :

« Tous nos opérateurs sont actuellement occupés, le délai d'attente est d'environ 20 minutes, merci de patienter. »

Naturellement, vous n'êtes pas satisfait, et vous vous demandez si vous aurez la patience d'attendre si longtemps, car vous devez absolument partir dans 15 minutes. Vous avez une idée, vous allez demander à un ami de prendre votre téléphone afin de vous libérer de cette attente.

Dans notre petite histoire, vous êtes une application et votre ami, un



thread de travail et votre opérateur, un service Web. Lorsque l'application appelle le service Web, dont la latence est extrêmement longue, l'application est figée et n'est plus disponible pour répondre à d'autres traitements. Dans ce cadre, l'utilisateur ne peut qu'attendre que le service se termine, ce qui n'est pas acceptable. Dans le cas où un thread de travail prend en main l'attente du service Web, l'utilisateur ne souffre plus d'une attente forcée ; l'application est disponible pour répondre à d'autres demandes. On note que le thread de travail n'a pas une grande valeur ajoutée. Il ne fait

qu'attendre le service. Dans les faits, ce thread ajoute un surcoût mémoire et une forme de complexité technique à votre application qui ne se justifie pas vraiment. Cependant, cette solution est largement utilisée, car elle est, à la fois simple à comprendre et facile à implémenter, mais entraîne des risques de concurrence entre le thread de travail et le thread principal de l'application.

Imaginons une nouvelle version de notre histoire. Cette fois, l'opérateur vous répond par le message suivant :

« Tous nos opérateurs sont actuellement occupés, merci de composer votre numéro de téléphone suivi d'un caractère '#', un de nos opérateurs vous rappellera dès que possible. »

Cette nouvelle version est bien plus intéressante, car elle libère l'application immédiatement après l'appel du service Web. Pour édifier un lien entre l'appel au service (qui ressemble plus à un enregistrement) un 'ticket' (dans notre exemple, le numéro de téléphone) permet de vous notifier lorsque le service web sera disponible. Vous n'avez plus besoin de créer un thread de travail dédié, un simple thread du pool de threads sera utilisé brièvement pour vous prévenir que le résultat du traitement est disponible.

Pour conclure nos deux histoires, je vous propose une synthèse sur l'opposition entre un appel synchrone et un appel asynchrone

Appel synchrone et Concurrency

- L'appelant gère la concurrence
- Démarre un autre thread qui attend la fin d'un appel à forte latence
- Pas besoin de callback/continuation

On qualifie ce type d'implémentation : « Asynchronisme au-dessus du

mode synchrone »

Ce type d'implémentation devrait être évité, car elle consomme des ressources inutilement

Appel asynchrone

- L'appelé gère la concurrence
- Pas d'utilisation de thread supplémentaire (si périphérique avec E/S)
- Pas d'attente pour l'appelant
- Besoin d'un callback/continuation

On qualifie ce type d'implémentation : « 100% asynchrone »

Ce type d'implémentation est parfait. Les avantages d'une implémentation 100% asynchrone sont nombreux. Premièrement, l'asynchronisme offre une meilleure combinaison efficacité/Mise à l'échelle (évite de bloquer des threads). Deuxièmement, le 100% asynchrone offre une solution bien plus facile à développer : l'appelant n'a pas à coder une couche de plomberie, l'appelant peut difficilement rencontrer des problèmes de threading, enfin il offre une forme d'abstraction vis-à-vis de la concurrence. Cependant une bonne implémentation ne présage pas d'une bonne API. En effet, il est parfaitement possible de proposer une implémentation 100% asynchrone, mais difficilement utilisable par un développeur.

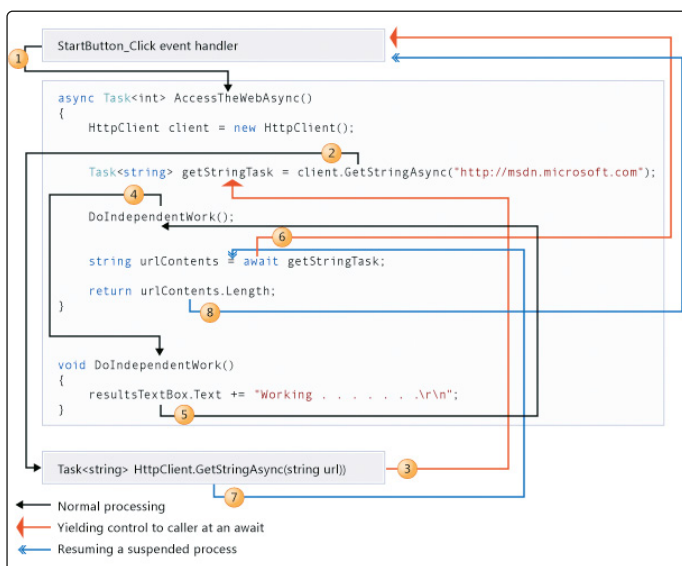
> Offrir une solution simple

Une bonne offre asynchrone devrait utiliser une implémentation 100% asynchrone. Cependant, entre la théorie et la pratique, il y a parfois un écart important, car, pour qu'une solution soit adoptée par la majorité des développeurs, il faut que celle-ci soit facile à utiliser. L'objectif dans ce chapitre est de se concentrer sur l'API et non sur l'implémentation.

> Code synchrone à transformer

Imaginons une solution asynchrone facile à utiliser en partant d'une simple méthode synchrone : `GetWebPage` dont l'intention est de retourner la page html d'un site web en fournissant l'URL de la page.

```
string GetWebPage (string uri)
{
    ...
}
```



```
...
}

void Test()
{
    string html = GetWebPage («http://blogs.msdn.com/b/pfxteam/»);
    Console.WriteLine (html);
}
```

> Première proposition

Comment rendre l'appel à `GetWebPage` asynchrone ? L'idée la plus simple serait sans doute d'utiliser le type générique `Action<TResult>` afin de passer une continuation à notre nouvelle API que nous appellerons `GetWebPageAsync`.

```
void GetWebPageAsync (string uri, Action<string> continuation)
{
    ...
    ...
}

void Test()
{
    GetWebPageAsync («http://blogs.msdn.com/b/pfxteam/», Console.
        WriteLine);
}
```

Cette première définition est sympathique, mais pas concluante si la méthode rencontre un problème. Cette API ne permet pas une prise en compte des erreurs et n'offre pas de contrôle sur le résultat.

> Seconde proposition

Si vous pratiquez déjà la programmation parallèle avec Visual Studio 2010, vous devez connaître le type `Task`. Dans notre exemple, nous allons profiter du type générique `Task<TResult>` pour retourner une information plus riche afin d'offrir plus de flexibilité à notre API. Vous pourriez faire une analogie rapide entre le type `Task` et le parallélisme, mais dans notre cas, nous réutilisons le type `Task` pour retourner une information et non pour paralléliser du code.

Dans Visual Studio 2012, le type `Task` unifie tous les traitements concurrents. Dans le vocabulaire multitâche, on parle de concurrence pour qualifier tout traitement utilisant un ou plusieurs threads. L'espace de la concurrence comprend plusieurs sous-domaines, par exemple le parallélisme.

Le type `Task` recouvre tous les domaines de la concurrence sous la forme traitement asynchrone. Il n'y a donc pas d'ambiguïté lorsque nous utilisons le type `Task` pour retourner le résultat du traitement asynchrone.

```
Task<string> GetWebPageAsync (string uri)
{
    ...
    ...
}
```

Le type générique `Task<TResult>` propose plusieurs propriétés intéressantes : `Result`, `Exception` et une méthode `ContinueWith()`. La propriété `Result` est synchrone et rend la main une fois le résultat

renseigné. La propriété `Exception` contient l'exception respective au problème rencontré si la méthode a échoué. Enfin, la méthode `ContinueWith` permet de passer une expression lambda qui sera exécutée dans un thread du pool de threads une fois le traitement terminé.

```
void Test()
{
    GetWebPageAsync("http://blogs.msdn.com/b/pfxteam/").Continue
    With(task =>
        Console.WriteLine (task.Result));
}
```

Cette dernière proposition, semble séduisante, car elle offre à la fois un excellent contrôle sur l'exécution du traitement et est simple d'utilisation.

> Code synchrone plus complexe à transformer

Imaginons maintenant un autre exemple synchrone un peu plus compliqué que le précédent, en introduisant une boucle autour de notre appel `GetWebPage` :

```
string GetWebPage (string uri)
{
    ...
    ...
}

void Test()
{
    for (int i = 0; i < 5; i++)
    {
        string html = GetWebPage("http://blogs.msdn.com/b/pfxteam/");
        Console.WriteLine (html);
    }
}
```

Cette version est bien plus compliquée à transformer en asynchrone. La boucle 'for' nous impose de nouvelles contraintes pour implémenter une API asynchrone. Une implémentation possible est d'utiliser la récursivité pour assurer l'ordre des appels à la méthode **GetWebPageAsync** asynchrone, afin de respecter le comportement de la version synchrone.

```
Task<string> GetWebPageAsync(string uri)
{
    ...
    ...
}

int _i = 0;

void Test()
{
    GetWebPageAsync("http://blogs.msdn.com/b/pfxteam/").Continue
    With (task =>
        {
            Console.WriteLine (task.Result);
        }
    );
}
```

```
if (++_i < 5)
    Test();
});
}
```

Pour maintenir notre algorithme, nous sommes obligés d'introduire une variable membre « `_i` », afin de maintenir l'état courant de la boucle. Cette implémentation illustre les limites d'un support de l'asynchronisme dans le cadre d'une structure de contrôle de type boucle. Cette implémentation n'est donc pas satisfaisante, car le code est trop complexe pour assurer une compréhension et une maintenance facile du code. On constate que la définition d'une API asynchrone à la fois simple et intégrable avec des structures de contrôle n'est pas une mince affaire.

Proposition avec C# 5.0

Pour conclure cette étude, nous illustrons notre exemple avec l'offre C# 5.0 :

```
Task<string> GetWebPageAsync (string uri)
{
    ...
    ...
}

async void Test()
{
    for (int i = 0; i < 5; i++)
    {
        string html = await GetWebPageAsync("http://blogs.msdn.com/
        b/pfxteam/");
        Console.WriteLine (html);
    }
}
```

Quelques explications sur l'utilisation de cette nouvelle offre : pour activer l'asynchronisme de C# 5.0, il vous faut utiliser de nouveaux identifiants `async` et `await` dans un contexte précis. Le mot `async` doit apparaître au début de la signature de la méthode (ou d'une expression lambda) contenant un ou plusieurs appels asynchrones utilisant le mot `await` : il précède l'appel d'une méthode asynchrone qui se doit de retourner une instance du type `Task` si la méthode asynchrone ne retourne rien, ou une instance de `Task<TResult>` si la méthode retourne un type `TResult`.

> Conclusion

Dans cet article nous avons posé le décor au travers de deux points : les motivations de l'asynchronisme et comment définir une interface utilisable. Ces deux points sont très importants, mais ne préjugent pas de l'implémentation sous-jacente. En effet, nous pouvons noter que la manière dont l'implémentation est construite n'a pas été détaillée dans cette première partie. Dans la seconde partie, nous plongerons dans l'offre asynchrone Microsoft avant C# 5.0.



Bruno Boucard
MVP – Expert en parallélisme
Architecte FINAXYS

Gatling, nouvelle solution libre pour les tests de montée en charge

Dans le domaine des outils de test de montée en charge permettant d'étudier les performances d'applications web, il existe globalement plusieurs catégories : des outils commerciaux pour la majorité très onéreux, des outils gratuits basiques ou austères, ou encore des outils gratuits devenus obsolètes avec le temps. Voici une nouvelle solution, Gatling, un outil de test de montée en charge gratuit et open source, s'appuyant sur un ensemble de technologies dont une des facultés est d'allier simplicité et fiabilité.

Nous allons dans cet article présenter l'outil sous la forme d'un didacticiel permettant de vous aider à le découvrir, mais aussi, à le prendre en main.

> Présentation de l'outil

L'outil tire son nom de la mitrailleuse créée par l'américain Richard Gatling, qui fut une arme fiable, puissante et facile à alimenter.

Afin de parvenir à « égaler » les qualités de son patronyme, cette solution se différencie des autres outils de test de charge en évitant d'utiliser un thread pour simuler un utilisateur virtuel. Gatling s'appuie pour cela sur le modèle d'acteurs d'Akka consistant à communiquer par échange de messages asynchrones. Cette architecture permet de simuler un nombre d'utilisateurs plus important à partir d'une seule machine. L'utilisation d'Akka a poussé le choix du langage Scala pour le développement de l'outil, mais également pour le langage des scripts de tests. Ces derniers reposent plutôt en réalité sur un langage dédié en Scala que l'on appelle également DSL (Domain Specific Language). Un ensemble d'API, que nous présenterons par la suite, permet de simuler les opérations réalisées par un utilisateur naviguant dans l'application web; le test consistant par exemple, à tester en émettant des requêtes HTTP.

Pour mettre en œuvre ces requêtes de manière optimale, Gatling s'appuie sur la librairie Asynchronous HTTP Client et sur le framework Netty qui sont également sous licence Apache 2.0. Enfin, pour permettre l'affichage de rapports de tests, l'outil génère un ensemble de pages HTML reposant sur les librairies javascript highstock et highcharts afin de produire des graphiques de manière optimisée. Maintenant que vous connaissez l'architecture technique et les différentes briques qui composent l'outil, passons aux explications pour l'installation.

> Installation de l'outil

Il existe principalement deux possibilités pour utiliser Gatling en tant qu'outil de test de montée en charge :

- Télécharger l'outil et dézipper l'archive dans un répertoire mettant à disposition deux scripts (bat ou shell selon votre OS) permettant de lan-

cer l'outil d'enregistrement d'un scénario puis de démarrer le moteur d'exécution d'un test

- Intégrer la solution au sein de l'outil de développement Eclipse, afin de disposer d'un projet « Gatling » permettant de manière plus confortable d'enregistrer, éditer (en bénéficiant notamment de la complétion) et exécuter les scripts de test

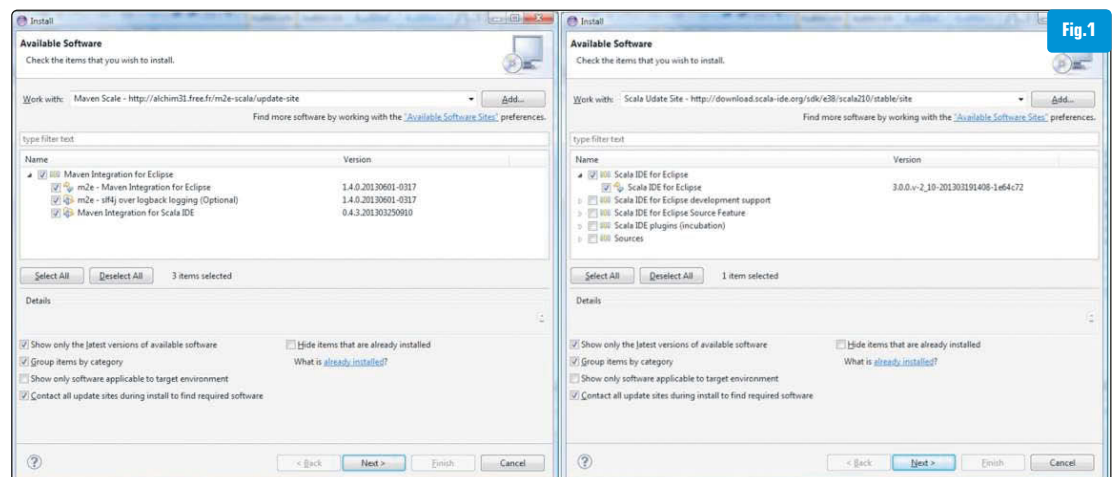
La première possibilité permet de déployer une solution de tir en quelques minutes alors que la seconde met à disposition un environnement de développement parfois nécessaire pour réaliser des scénarios de tests complexes. Nous allons détailler l'utilisation au sein d'Eclipse qui nous paraît être la meilleure façon d'utiliser pleinement l'outil. Dans tous les cas, il vous faut disposer d'un JDK 6 minimum, sachant qu'il est préconisé d'utiliser la version 7.

Il faut commencer par procéder à l'installation de l'outil de développement Eclipse dont la version supportée est Juno. L'ajout de plusieurs plug-in [Fig.1] est nécessaire. Voici les liens à ajouter :

- Plug-in Scala IDE pour Eclipse : <http://download.scala-ide.org/sdk/e38/scala210/stable/site>
- Plug-in Maven Scala pour Eclipse : <http://alchim31.free.fr/m2e-scala/update-site>

Par la suite [Fig.2], il faut configurer le catalogue archetype Maven suivant : <http://repository.exciys.com/content/groups/public/archetype-catalog.xml> Voilà, vous pouvez maintenant créer un nouveau projet Maven en choisissant l'archetype Gatling qui vous propose à ce jour deux versions de l'application : 1.5.2 et 2.0.0. Dans notre cas, nous allons travailler avec la version actuellement la plus récente. Il faut ensuite renseigner un groupId et un artifactId pour votre projet [Fig.3]

Votre projet est automatiquement créé dont voici l'arborescence et les différents fichiers qui le composent [Fig.4].



Ce projet comprend notamment :

- Recorder.scala : programme Scala permettant de lancer l'outil enregistreur appelé également « recorder » en anglais
- Engine.scala : programme Scala représentant le moteur permettant d'exécuter un script de test
- Le répertoire « data » permettant de stocker des fichiers à charger pour simuler des jeux de données
- Plusieurs fichiers de configuration pour l'outil d'enregistrement, le moteur, etc.
- Un fichier logback.xml permettant d'être plus ou moins évocateur pour chaque requête HTTP émise. Il est cependant conseillé d'éviter de mettre le niveau de trace le plus détaillé lors de l'exécution des tests de montée en charge pour ne pas biaiser les résultats.

Si vous souhaitez changer de version suite à la mise à disposition d'une nouvelle fonctionnalité qui vous intéresse ou d'une correction de bug, il vous suffit d'éditer le fichier pom.xml et de renseigner la version de ces librairies :

```
<gatling.version>2.0.0-M3a</gatling.version>
<gatling-highcharts.version>2.0.0-M3a</gatling-highcharts.version>
```

Pour connaître les différentes versions, vous pouvez vous rendre sur le github du projet et voir les versions des différents tags mis à disposition : <https://github.com/excilys/gatling>

Vous l'aurez compris, nous allons vous montrer comment fonctionne la version 2.0.0-M3a de Gatling.

> Enregistrement d'un script de test

Pour faciliter la mise en place d'un script de test, l'outil propose un « recorder » permettant d'enregistrer les requêtes HTTP d'un scénario réalisé par l'intermédiaire d'un navigateur. Pour lancer le recorder à partir d'Eclipse, il suffit de cliquer sur le fichier « Recorder.scala » et d'exécuter en tant qu'application Scala [Fig.5].

Il existe deux modes d'enregistrement :

- HTTP Proxy : le recorder joue un rôle de proxy par lequel votre navigateur doit pointer afin d'enregistrer l'ensemble des requêtes. Par défaut, le port HTTP est 8000 mais cela est configurable
- HAR Converter : il est envisageable d'enregistrer votre scénario avec votre navigateur préféré et de générer un fichier au format HAR (HTTP ARchive) représentant l'ensemble des requêtes (il s'agit d'une norme de description au format JSON). Ce fichier peut alors être chargé dans l'outil qui le transforme en script

Dans les deux cas, il est possible d'indiquer une stratégie d'enregistrement en créant des filtres permettant de spécifier les ressources à exclure (comme par exemple les éléments statiques css, js, etc.) ou bien les ressources uniquement à enregistrer (comme par exemple /*.php) selon les besoins des tests.

Lors de l'enregistrement en mode HTTP proxy, il est possible de saisir un tag entre chaque page afin de voir apparaître des commentaires dans le script généré. Par ailleurs, l'outil présente les informations sur les requêtes et les réponses [Fig.6].

Un script très propre et facilement compréhensible est automatiquement généré intégrant notamment les temps de pause réels pris entre chaque clic lors de l'enregistrement :

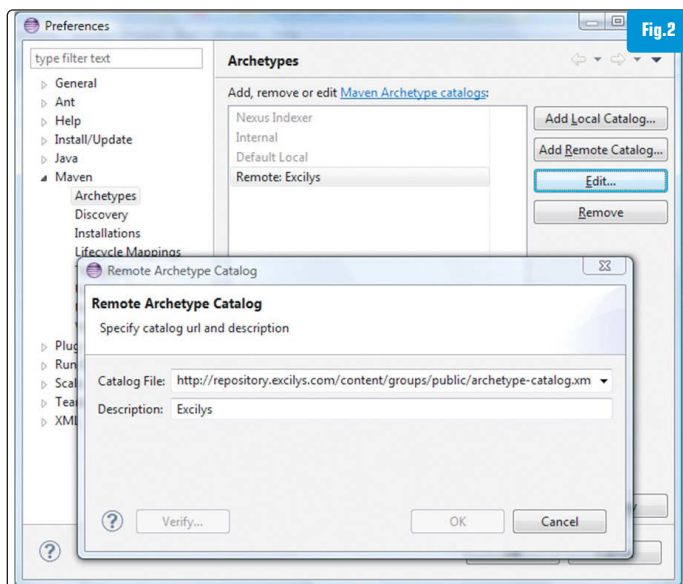


Fig.2

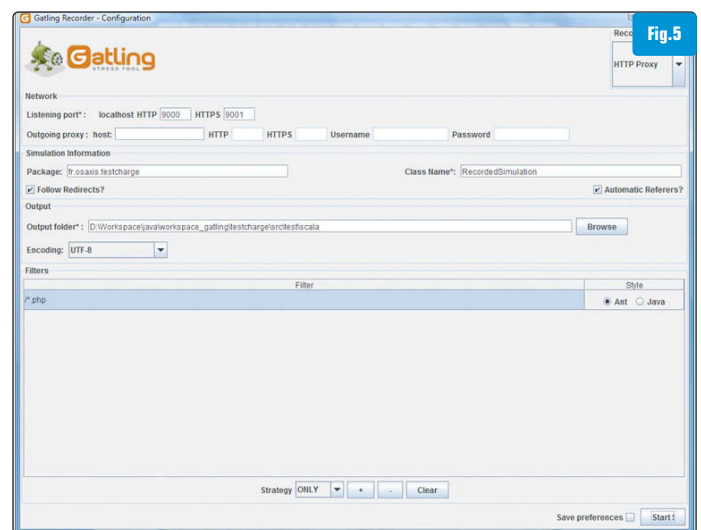


Fig.5

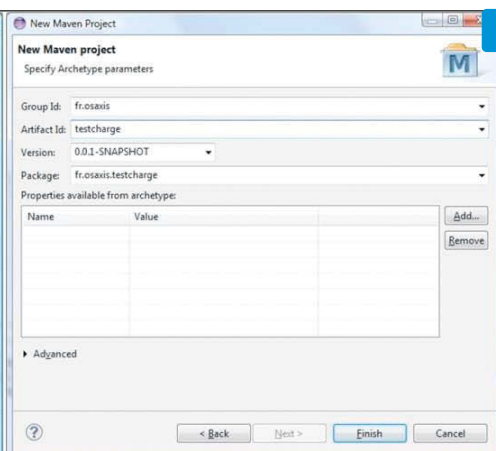
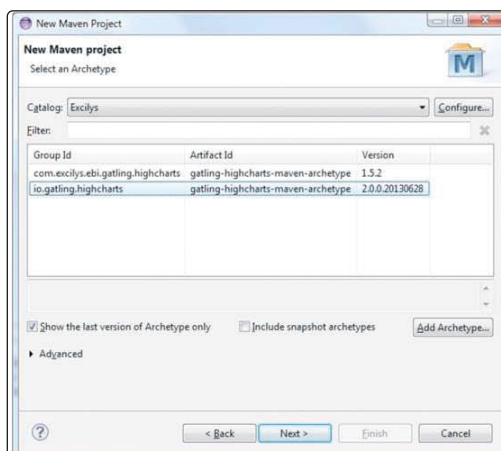


Fig.3

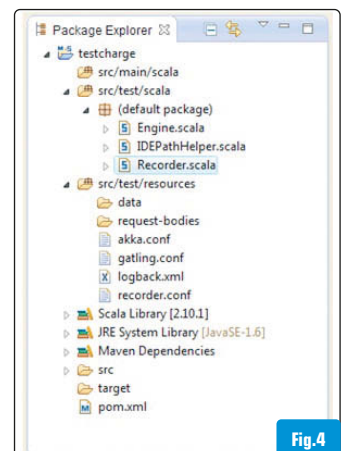


Fig.4

```
class RecordedSimulation extends Simulation {

val httpProtocol = http
    .baseUrl("http://osaxis.fr")
    .acceptHeader("text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8")
    .acceptEncodingHeader("gzip,deflate,sdch")
    .acceptLanguageHeader("fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4")
    .userAgentHeader("Mozilla/5.0 (Windows NT 6.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1500.72 Safari/537.36")

val scn = scenario("Scenario Name")
    //accueil
    .exec(http("request_1")
        .get(""/index.php""))
    //métiers
    .pause(7)
    .exec(http("request_2")
        .get(""/accompagner.php""))
    //références
    .pause(10)
    .exec(http("request_3")
        .get(""/projets.php""))

setUp(scn.inject(atOnce(1 user))).protocols(httpProtocol)
}
```

> Personnalisation du script de test

A partir du fichier généré, on peut faire évoluer le script afin de le rendre plus dynamique en injectant des jeux de données variables. Il est également possible de modifier le script en analysant les réponses des requêtes afin d'y extraire des informations nécessaires pour poursuivre correctement le scénario.

Le langage DSL proposé pour ces scripts dispose de nombreuses API segmentées selon sept grandes catégories que nous vous présentons à travers quelques exemples. A noter que nous présentons les principales API mais elles ne sont pas toutes décrites ici.

> Scenario definition

Cette première catégorie permet de déclarer un ou même plusieurs scénarios dans un script à partir de l'API « scenario » afin de repro-

duire le comportement d'utilisateurs. L'autre API importante est « exec » permettant de réaliser des traitements comme par exemple l'exécution d'une requête HTTP :

```
val scn = scenario("Scenario1").exec(http("request_1").get(""/index.php""))
val scn2 = scenario("Scenario2").exec(http("request_2").get(""/index2.php""))
```

Une notion de « Group » offre la possibilité de regrouper plusieurs requêtes afin d'avoir des statistiques pour un ensemble d'enchaînements, sachant que l'on peut imbriquer plusieurs groupes. De nombreuses autres API telles que « repeat » et « during » permettent d'itérer une partie d'un scénario, alors que l'insertion des temporisations au sein du scénario se fait à l'aide de « pause » et « pauseExp ». Enfin, il est possible de réaliser des conditions « doIf » et « doIfOrElse » en s'appuyant sur des attributs en session accessibles à partir d'EL (Expression Language).

> HTTP Action

Comme vous avez pu le constater dans l'exemple de code précédent, une requête HTTP de type GET a été réalisée mais il est bien évidemment possible de simuler d'autres types de requêtes HTTP comme « POST », « PUT », « DELETE », etc. et également de passer des paramètres à ces requêtes. Voici par exemple une requête en POST pour l'authentification à un site en HTTP :

```
val scn = scenario("SimpleLogin")
    .exec(http("login")
        .post(""/login.php""))
        .param("login", "john")
        .param("password", "azerty"))
```

Une API est disponible pour gérer les authentifications basiques :

```
val scn = scenario("BasicLogin")
    .exec(http("login")
        .get(""/login.php""))
        .basicAuth("john", "azerty"))
```

> HTTP Configuration

Ce bloc de code est présent au début du script généré suite à l'enregistrement avec le recorder :

```
val httpProtocol = http
    .baseUrl("http://osaxis.fr")
    .acceptHeader("text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8")
    .acceptEncodingHeader("gzip,deflate,sdch")
    .acceptLanguageHeader("fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4")
    .userAgentHeader("Mozilla/5.0 (Windows NT 6.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1500.72 Safari/537.36")
```

Ceci permet de configurer seulement une fois un certain nombre d'informations pour l'ensemble des requêtes HTTP qui seront injectées par la suite dans le script. Il est possible par exemple de spécifier l'URL de base du site à tester, sachant qu'il est même proposé de renseigner plusieurs « baseURLs » lorsque votre site est en cluster et que vous souhaitez injecter une charge sur chacun des serveurs sans passer par le répartiteur de charge. Il est également envisageable de configurer un proxy HTTP si vous êtes contraint de passer par un proxy pour accéder au site à tester.

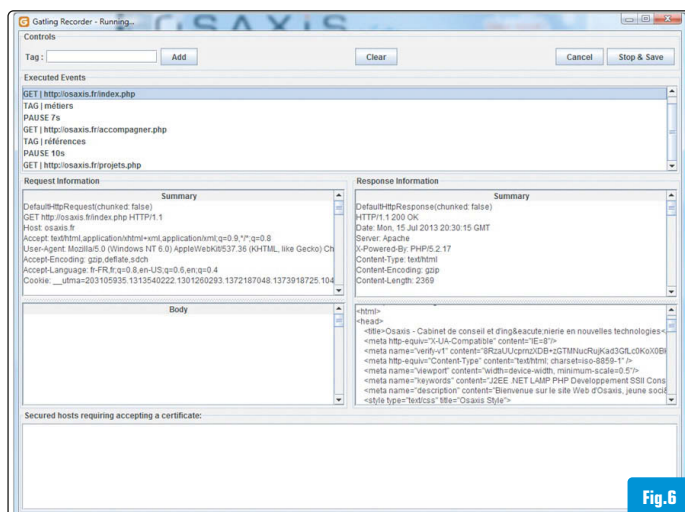


Fig.6

> Feeder definition

Pour pouvoir injecter des jeux de données variables, vous pouvez charger des fichiers de différents formats voire même récupérer des informations en base de données. Plusieurs stratégies offrent la possibilité de manipuler des données afin de les injecter de manière aléatoire ou bien circulaire. Reprenons notre exemple précédent mais en utilisant désormais le fichier « users.csv » placé dans le répertoire « data » de votre arborescence et dont voici le contenu :

```
login,password // la première ligne permet d'attribuer un nom à chaque donnée
john,azerty
kevin,soleil
```

Le code pour le charger et l'injecter dans le scénario est le suivant :

```
val users = csv("users.csv") // chargement du fichier

val scn = scenario("FeedLogin")
    .feed(users) // injection des données
    .exec(http("login")
        .post(""/login.php"")
        .param("login", "${login}") // passage du login
        .param("password", "${password}")
    // passage du password
    .check(currentLocation.is(baseUrl+"/private.php")))
```

> Checks

Comme vous avez pu le constater dans l'exemple précédent, nous avons vérifié que la page obtenue après l'authentification était bien « private.php » en utilisant l'API « check ». Il existe d'autres possibilités, comme par exemple de vérifier le code retour HTTP qui est en principe de « 200 » si tout va bien. Il est envisageable de valider le contenu retourné mais également de l'analyser (de le parser) afin de récupérer des informations contenues dans le flux en utilisant des expressions régulières. Ceci est indispensable dans certains cas de figure où un identifiant est généré dynamiquement dans la page et qu'il faut ensuite le transmettre en paramètre de la requête suivante. Voici un exemple :

```
val scn = scenario("UpdateScore")
    .feed(users)
    .exec(http("login")
        .post(""/login.php"")
        .param("login", "${login}")
        .param("password", "${password}")
        .check(currentLocation.is(baseUrl+"/private.php")))

    .exec(http("monde")
        .get(""/private_monde.php?monde=3"")
        .check(
            regex("""<input type="hidden" name="tab"
value="([0-9]*)">""") // expression régulière
            .saveAs("idtab"))) // sauvegarde d'une variable

    .exec(http("update")
        .post(""/updatescore.php"")
        .headers(headers_xml)
        .param("tab", "${idtab}") // utilisation de la variable
        .param("score", "31000")
        .param("birds", "2")
        .check(status.is(200))) // vérification du status code 200
```

> Scenario configuration

Ce bloc de code est présent à la fin du script généré suite à l'enregistrement avec le recorder :

```
setUp(scn.inject(atOnce(1 user))).protocols(httpProtocol)
```

Cette ligne de code permet de configurer le scénario de test à simuler en précisant le nombre d'utilisateurs. Par défaut, c'est simplement un utilisateur virtuel qui est proposé par l'enregistreur mais en général, c'est une montée en charge progressive qui est requise. Celle-ci se fait en précisant notamment des durées de préparation. Voici une configuration de scénario s'inspirant fortement de l'exemple fourni dans le wiki de l'outil et qui permet d'avoir une idée du spectre des possibilités offertes par la solution :

```
setUp(scn.inject(nothingFor(4 seconds),
    atOnce(10 users),
    ramp(10 users) over (5 seconds),
    constantRate(20 usersPerSec) during (15 seconds),
    rampRate(10 usersPerSec) to(20 usersPerSec) during(10 minutes),
    split(1000 users).into(ramp(10 users) over (10 seconds))
    .separatedBy(10 seconds),
    split(1000 users).into(ramp(10 users) over (10 seconds))
    .separatedBy(atOnce(30 users)))).protocols(httpProtocol)
```

> Assertions

Afin de valider que le scénario répond à certaines exigences, l'outil permet de spécifier un certain nombre de vérifications sur les temps de réponse ou le nombre de requêtes en échec. Cela peut être au niveau global ou bien sur une requête en particulier.

Voici un exemple permettant de vérifier que le temps global de chaque requête est inférieur à une seconde, qu'il y a plus de 98% de requêtes réussies et que la durée de traitement de la page de login est comprise entre 100 et 500 millisecondes :

```
setUp(scn.inject(atOnce(1 user))).protocols(httpProtocol)
    .assertions(
        global.responseTime.max.lessThan(1000),
        global.successfulRequests.percent.greaterThan(98),
        details("login").requestsPerSec.greaterThan(100).lessThan(500))
```

Suite à l'exécution du test, un rapport indique pour chaque assertion si elle est vérifiée ou non et si la simulation a échoué ou réussi selon vos critères :

```
Global : max response time is less than 1000 : false
Global percentage of requests OK is greater than 98 : true
login : requests per second is greater than 100 : false
login : requests per second is less than 500 : true
Simulation failed.
```

> Exécution et analyse des résultats

Maintenant que vous avez un script de test permettant de parfaitement simuler le ou les scénarios avec le nombre d'utilisateurs de votre choix, il faut désormais lancer le test (en pratique, le test est exécuté au cours de la création du script pour valider le fonctionnement du scénario surtout lorsqu'il est complexe).

Pour lancer votre test depuis Eclipse, il suffit de cliquer sur le fichier « Engine.scala » et de l'exécuter en tant qu'application Scala. Si votre

projet comporte plusieurs scripts de test, la liste apparaît dans la console et vous pouvez choisir la simulation de votre choix. Ensuite, l'outil vous permet de spécifier un identifiant (afin de pouvoir facilement identifier la simulation réalisée parmi l'ensemble des simulations que vous pouvez être amené à produire) et une description (permettant d'avoir un commentaire dans le rapport).

Pendant l'exécution des tests, des données apparaissent toutes les cinq secondes permettant d'avoir un aperçu de l'état du test en cours [Fig.7]. A la fin du test, un rapport textuel simple résume le test qui a été réalisé [Fig.8].

Par ailleurs, un certain nombre de fichiers sont automatiquement générés dans le répertoire « target/results/id-timestamp » de votre projet dans Eclipse (il faut rafraîchir votre projet pour les voir apparaître). Un fichier « simulation.log » contenant des informations sur chaque requête émise est généré. Il est envisageable d'indiquer des informations personnalisées dans ce fichier en ajoutant à la configuration HTTP l'API « extraInfoExtractor » permettant de récupérer des données dans la « session », la « request », la « response » ou bien le « status » :

```
val httpProtocol = http
    .baseUrl("http://www.osaxis.fr")
    .acceptHeader("*/")
    .acceptEncodingHeader("gzip, deflate, sdch")
    .acceptLanguageHeader("fr-FR, fr;q=0.8, en-US;q=0.6, en;q=0.4")
    .extraInfoExtractor((status: Status, session: Session, request: Request, response: Response)
        => { List[String](request.getRawUrl(),
            response.getStatusCode.toString())})
```

Un fichier « stats.tsv » comporte un ensemble de statistiques sur la simulation réalisée comme par exemple le nombre de requêtes, le nombre de requêtes en erreur, le temps de réponse minimum, le temps de réponse maximum, etc.

Un rapport complet au format HTML est également généré contenant un grand nombre de tableaux et graphiques agréables à l'œil, et surtout, représentant une mine d'informations sur la simulation réalisée [Fig.9].

> Conclusion

Gatling dispose des fonctionnalités nécessaires pour mener à bien une campagne de tests de montée en charge. On ressent que cet outil a été conçu par des développeurs qui connaissent bien les besoins, et pour des développeurs qui veulent une solution simple, flexible et agréable. Les puristes diront qu'il manque une interface pour faciliter l'exécution et la configuration de la simulation.

On peut regretter l'absence d'une solution pour coordonner une simulation depuis plusieurs injecteurs (dans le cas où une très grosse charge est à réaliser), mais cela reste malgré tout envisageable

en réalisant quelques opérations manuelles indiquées sur le wiki de l'outil. Ceci dit, pour une solution gratuite et encore jeune (à peine deux ans), l'outil se révèle déjà complet et bien documenté. Par ailleurs, une véritable communauté semble s'être créée et le support apparaît réactif.

Sachez que nous avons essentiellement présenté l'outil comme étant une solution de test de montée en charge, mais il est tout à fait envisageable d'utiliser les scripts pour réaliser des tests fonctionnels de non régression. En effet, il existe également le plug-in Jenkins permettant de disposer d'un rapport de tests pour chaque construction au sein de votre usine logicielle.

Liens et références

Site de l'outil : <http://gatling-tool.org>

Documentation de l'outil : <https://github.com/excilys/gatling/wiki>

Google User Group de l'outil :

<https://groups.google.com/forum/#!forum/gatling>

Modèle d'acteurs Akka : <http://akka.io>

Langage Scala : <http://www.scala-lang.org>

Netty : <http://netty.io>

Async HTTP Client : <https://github.com/AsyncHttpClient/async-http-client>

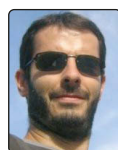
Librairies javascript highstock et highcharts :

<http://www.highcharts.com/products/highcharts> et

<http://www.highcharts.com/products/highstock>

Licence Apache 2.0 : [http://www.tldrlegal.com/license/apache-license-2.0-\(apache-2.0\)](http://www.tldrlegal.com/license/apache-license-2.0-(apache-2.0))

Eclipse Juno : <http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops4/R-4.2.2-201302041200/eclipse-SDK-4.2.2-win32.zip>



Jean-Sébastien Mercy,
consultant Osis.



```
=====
2013-07-23 21:35:42                               25s elapsed
----- UpdateScore -----
[=====] 70%
waiting: 0 / running: 3 / done: 7
----- Requests -----
> Global (OK=45 KO=6 )
> login (OK=8 KO=2 )
> monde (OK=8 KO=2 )
> logout (OK=8 KO=2 )
> login Redirect 1 (OK=8 KO=0 )
> update (OK=8 KO=0 )
> logout Redirect 1 (OK=5 KO=0 )
=====
```

```
----- Global Information -----
> numberOfRequests 54 (OK=48 KO=6 )
> minResponseTime 100 (OK=100 KO=120 )
> maxResponseTime 10760 (OK=10760 KO=260 )
> meanResponseTime 3085 (OK=3449 KO=173 )
> stdDeviation 3590 (OK=3648 KO=62 )
> percentiles1 9750 (OK=9790 KO=260 )
> percentiles2 9820 (OK=10760 KO=260 )
> meanNumberOfRequestsPerSecond 3 (OK=2 KO=0 )
----- Response Time Distribution -----
> t < 800 ms 18 ( 33%)
> 800 ms < t < 1200 ms 5 ( 9%)
> t > 1200 ms 25 ( 46%)
> failed 6 ( 11%)
=====
```


jours un objet de type **CloudMediaContext**. Son constructeur prend en paramètre le nom du service ainsi que la clé d'accès primaire récupérée sur le portail d'administration.

```
CloudMediaContext mediaContext = new CloudMediaContext("infinitevod",
    "<VOTRE CLE>");
```

Azure Media Services utilise une notion « d'asset » pour manipuler les différents fichiers, que ce soit en entrée d'une tâche (un encodage par exemple) ou en sortie. Pour commencer, il est nécessaire de créer un asset et lui associer un fichier (blob) avec lequel travailler.

```
//création de l'asset et déclaration du fichier lié
IAsset monAsset = mediaContext.Assets.Create("monAsset", Asset
CreationOptions.None);
IAssetFile monFichier = monAsset.AssetFiles.Create("Wildlife2.wmv");

//upload du fichier
monFichier.Upload(@"C:\Wildlife2.wmv");
```

Pour pouvoir lancer l'encodage de ce fichier, il faut récupérer une instance de **IMediaProcessor**, qui correspond à un encodeur / packager fourni par Microsoft ou un partenaire tiers. Dans le cas présent, c'est le *Windows Azure Media Encoder* qui est utilisé. Il n'est rien d'autre qu'un portage de l'encodeur de Microsoft Expression porté dans le Cloud. Il est également nécessaire de préciser une configuration (preset) pour la tâche d'encodage, de manière à ce qu'elle sache quel travail réaliser : concrètement il s'agit d'indiquer à l'encodeur le format de sortie que vous souhaitez obtenir. Il est possible de récupérer la liste des configurations par défaut disponibles sur MSDN ou de créer sa propre configuration (un fichier XML).

```
//définition du preset
string preset = "H264 Broadband 720p";

//récupération du media processor
IMediaProcessor mediaProcessor = mediaContext.MediaProcessors
.Where(f => f.Name == "Windows Azure Media Encoder").AsEnumerable
().FirstOrDefault();
```

Il est alors possible de créer la tâche d'encodage au sein d'un job Azure Media Services :

```
//création du job
IJob job = mediaContext.Jobs.Create("monJob");

//création de la tâche
ITask task = job.Tasks.AddNew("maTache", mediaProcessor, preset,
TaskOptions.None);
```

La prochaine étape consiste à indiquer à la tâche l'asset avec lequel elle doit travailler, mais également l'asset qu'elle doit produire en sortie.

```
//ajout de l'asset en entrée
task.InputAssets.Add(monAsset);
```

```
//récupération de l'asset de sortie
IAsset monAssetDeSortie = task.OutputAssets.AddNew("monAssetDeSortie");
```

Il ne reste qu'à soumettre le job à Azure Media Services et patienter tant que celui-ci est en cours d'exécution. A tout moment, il est possible de connaître l'état du job via sa propriété **State**.

```
//soumission du job
job.Submit();

//attente du job
Task jobTask = job.GetExecutionProgressTask(CancellationToken.None);
jobTask.Wait();

//vérification de l'état
if (job.State == JobState.Finished)
{
    Console.WriteLine("job terminé !");
}
```

Le job étant terminé, il est possible de visualiser le résultat directement depuis le portail d'administration Azure, dans la section « Contenu » du service de média. A présent, il est possible de récupérer et publier l'asset de sortie de deux manières : soit en cliquant sur le bouton « Publier » après avoir sélectionné l'élément dans la section contenu du service de médias, soit par le code.

Pour réaliser cette opération par le code, il faut tout d'abord créer une politique d'accès au fichier, représentée par l'interface **IAccessPolicy**, ainsi qu'un objet de type **ILocator**, qui peut être soit de type **SAS** (basé sur les Shared Access Key des blobs) pour du téléchargement direct, soit *Origine* pour du streaming adaptif (ce type de locator implique la réservation d'unités de streaming dédiées dans la section Origine du service de média).

```
//création de la politique d'accès
IAccessPolicy accessPolicy = mediaContext.AccessPolicies.Create
("Lecture seule 42 jours", TimeSpan.FromDays(42), AccessPermissions
.Read);

//création du locator
ILocator locator = mediaContext.Locators.CreateSasLocator(job.
OutputMediaAssets[0], accessPolicy);
```

Enfin, il ne reste plus qu'à utiliser le locator pour construire l'url d'accès au fichier, comme suivant :

```
//création de l'url d'accès
string url = string.Format("{0}/{1}{2}", locator.BaseUri, job.
OutputMediaAssets[0].AssetFiles.First().Name, locator.Content
AccessComponent);
```

> Conclusion

Cet article a permis d'avoir un aperçu de la brique Azure Media Services, tant sur la partie des enjeux fonctionnels, métiers et financiers auxquels elle permet de répondre, que sur la partie purement technique et prise en main du SDK Windows Azure Media Services. A vos médias !

Julien Corioland

Tech Lead Windows Azure & Web, Infinite Square

<http://blogs.infinite-square.com/b/beedoo>

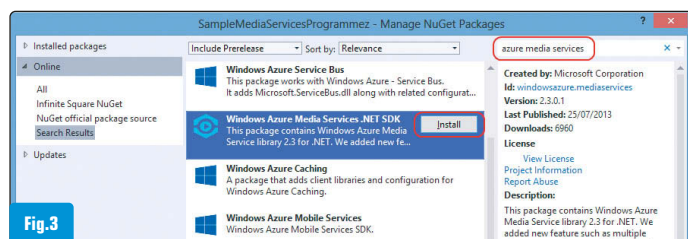


Fig.3

Protéger son site avec une Web Application Firewall (WAF)

A l'heure où les netbots lancent leurs attaques automatisées en masse à la moindre opportunité et où n'importe quel "script kid" peut mettre à mal un serveur en faisant tourner quelques scripts metasploit populaires, protéger son site avec un WAF n'a rien de superflu. Cela tombe bien, cet article vous explique pourquoi et comment avec ModSecurity et Naxsi.

> Pourquoi mettre en place un WAF ?

Pour comprendre le principe du WAF, revenons au concept de "firewall applicatif" : là où un pare-feu classique (comme netfilter/iptables) vous propose de filtrer vos entrées/sorties au niveau de la couche réseau, un pare-feu applicatif filtre, comme son nom l'indique, au niveau applicatif. Son rôle sera donc d'autoriser et de refuser une connexion à un service spécifique en fonction des différents paramètres de celle-ci : dans le cas d'un WAF, on pourra donc filtrer sur la forme des URL, les paramètres GET/POST, les IP clientes, les user-agents, etc..

Prenons un exemple concret. Un vilain pirate souhaite profiter d'une faille de type injection SQL pour rentrer sur le module d'administration de votre site. Il saisit donc dans votre formulaire de connexion un mot de passe du genre :

```
' or '1'='1'
```

... pour qu'une requête SQL de contrôle de login du type :

```
SELECT * from utilisateurs where nom = 'NOM_SAISI' and password = 'PASSWORD_SAISI'
```

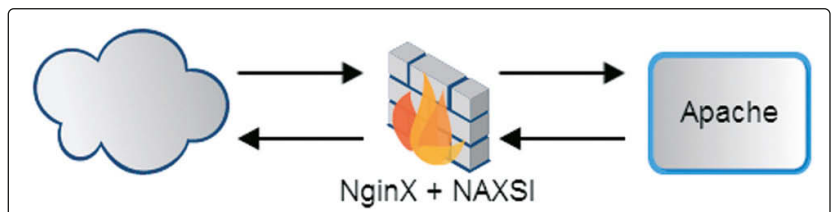
se transforme en :

```
SELECT * from utilisateurs where nom = 'admin' and password = '' or '1'='1'
```

et donne accès à coup sûr au pirate ! Si vous avez déjà été confronté à ce genre de chose, vous savez qu'il n'y a en principe qu'une seule vraie solution : avoir un site bien codé qui contrôle systématiquement les données saisies par l'utilisateur.

Oui mais voilà, il est possible que vous ne soyez pas l'auteur du code du site, ou que vous ayez utilisé différents modules open-source pouvant faire l'objet de failles encore non découvertes. L'audit de code approfondi que nécessiteraient ces situations est malheureusement hors de portée de la plupart des webmasters faute de temps et de moyens à y consacrer.

C'est justement là qu'intervient le WAF : en filtrant toutes les requêtes, il pourra refuser celle de notre pirate qui utilise dans notre exemple simplifié à l'extrême, un motif d'attaque très fréquent et aisément repérable. Votre site équipé d'un WAF pourra ainsi ren-



Exemple d'architecture Nginx/Naxsi en reverse proxy

voyer une erreur 403 au pirate avant même que votre code serveur n'ait été appelé.

Ce qui fait toute la puissance de l'outil, c'est qu'il peut boucher quantité de failles de votre site sans toucher à une ligne de son code... et sans même d'ailleurs que vous soyez conscient de ces failles ! Attention cependant, si un WAF bien paramétré renforce considérablement la sécurité d'un site, il n'en fait pas une assurance tous risques. Les bonnes pratiques de sécurité web de base, comme décrites par exemple dans le top 10 de l'OWASP, doivent rester une exigence pour les développeurs web. Mais passons maintenant aux choses sérieuses en regardant comment mettre en oeuvre deux outils open-source de référence.

> ModSecurity

ModSecurity est probablement aujourd'hui la solution WAF la plus déployée dans le monde. A la base, module destiné à renforcer la sécurité des sites servis par Apache, ModSecurity peut désormais être aussi installé sur IIS (version 7 et supérieures) et NginX. Nous supposons pour la suite de cet article que nous allons configurer ModSecurity pour Apache mais l'essentiel des configurations est identique sur les autres serveurs web.

Une fois le module installé (des packages sont disponibles pour les principales distributions Linux, ainsi qu'un installateur windows pour la version IIS) et configuré pour fonctionner sur votre environnement, nous pouvons passer à la mise en place des règles qui vont renforcer la sécurité de vos sites.

Le plus simple pour débuter est d'appliquer les règles par défaut proposées par ModSecurity dans le fichier modsecurity.conf-recommended. D'une part ce fichier vous donnera une première base sur laquelle vous appuyer, d'autre part grâce à sa directive "SecRuleEngine DetectionOnly", il vous permettra de tester le déclenchement de vos règles dans un fichier de log sans rejeter les requêtes correspondantes. Une fois que vous aurez suffisamment de certitudes sur le fait que votre jeu de règles ne déclenche pas de "faux-positifs" (rejeter des requêtes normalement autorisées), vous pourrez basculer sur cette directive sur "SecRuleEngine On".

Mettons maintenant en place quelques règles personnalisées. Si par exemple, je souhaite empêcher l'attaque décrite plus haut, je pourrai définir une règle empêchant ce type de saisie :

```
SecRule ARG ' or '1'='1 "id:1"
```

Explications :

- SecRule définit une règle de filtrage
- ARG ' or '1'='1 vérifie si les paramètres de la requêtes HTTP GET ou POST contient la chaîne ' or '1'='1
- "id:1" définit un identifiant pour notre règle

En vérité cette règle serait peu utile dans la mesure où elle ne filtrerait qu'un type d'attaque possible parmi les dizaines que notre formulaire mal sécurisé permet. Nous pourrions passer par des expressions régulières pour filtrer plus de cas, mais nous allons utiliser une règle de ModSecurity assez bluffante :

```
SecRule ARG "@detectsqli" "id:2"
```

L'expression @detectsqli va utiliser une collection de patterns de détection d'injection SQL interne à ModSecurity... vous évitant ainsi de réinventer la roue pour gérer un problème maintenant bien connu. Notons au passage que cette simple ligne protège votre site sur tous les arguments pouvant être envoyés sur toutes les requêtes possibles de chacune de vos pages. C'est donc particulièrement puissant... au point que cela pourrait risquer de générer quelques faux-positifs. Imaginons par exemple que votre site contienne un forum dédié au développement web : dès qu'un membre postera un message contenant un peu de SQL, sa requête a de grandes chances de se voir filtrée. Pour contourner le problème, nous pouvons créer certaines zones non filtrées par notre règle. Dans le fichier whitelist.conf on écrira par exemple :

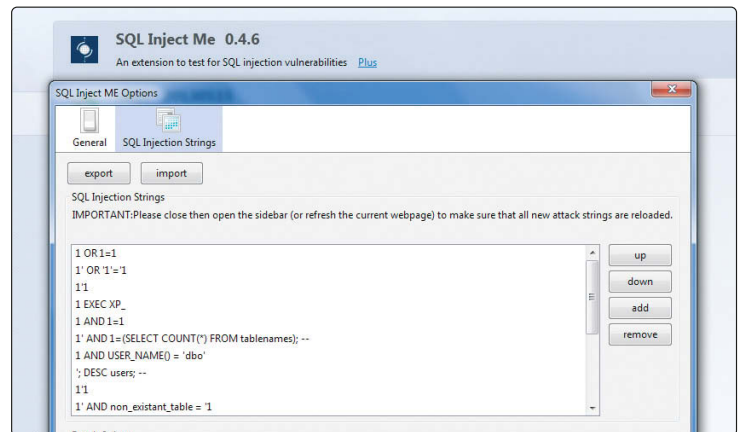
```
<LocationMatch "monforum/post.php">
SecRuleRemoveById 2 #suppression de la SecRule 2 pour le forum
</LocationMatch>
```

> Le Core Rules Set

Mais au-delà du sur-mesure, ne pourrait-on pas mettre en place un ensemble de règles génériques pouvant améliorer la protection de n'importe quel site ? Et bien si ! C'est possible grâce au Core Rules Set (CRS) proposé par l'OWASP. Le CRS librement téléchargeable, tire parti de l'expérience emmagasinée sur ModSecurity pour offrir des techniques de filtrage qui vont bien au-delà de la simple détection de pattern d'attaque évoquée plus haut. On peut citer par exemple :

- détection des violations du protocole HTTP
- utilisation d'un système de réputation d'IP
- détection des contenus web malicieux grâce l'API "Safe Browsing" de Google
- détection des attaques de type déni de service
- détection des robots
- intégration d'antivirus pour contrôle systématique des uploads de fichier

Pour installer le CRS, il vous suffit de créer un dossier modsecurity_crs dans le dossier de configuration d'Apache et d'y copier les fichiers téléchargés depuis <https://github.com/SpiderLabs/owasp-modsecurity-crs>



Quelques injections classiques paramétrées dans SQLInjectMe

Vous n'avez plus qu'à ajouter dans httpd.conf :

```
<IfModule security2_module>
Include conf/modsecurity_crs/*.conf
Include conf/modsecurity_crs/base_rules/*.conf
</IfModule>
```

... et à redémarrer Apache. Vous pouvez ensuite vous amuser à simuler des attaques (par exemple avec le plugin Firefox sql inject me) et à examiner vos logs pour vérifier ce que cela donne.

> Naxsi

Pour finir cette introduction aux WAF, examinons rapidement une alternative à ModSecurity intéressante à plusieurs titres : c'est une innovation française relativement récente (2011) soutenue par l'OWASP qui a choisi une approche volontairement très différente. Là où ModSecurity permet de mettre en place de nombreuses règles s'appuyant sur de multiples paramètres, Naxsi applique une stratégie simpliste : il donne un score aux requêtes HTTP en fonction d'un nombre de règles très limitées... si le score dépasse un certain seuil, la requête est rejetée. Problème de cette solution, elle génère beaucoup de faux-positifs : Naxsi propose donc un système de génération de liste blanche qui va lui permettre d'apprendre quelles sont les requêtes légitimes. Avantage considérable de cette stratégie comparativement à ModSecurity : un impact très faible sur les performances de votre site.

L'installation de Naxsi est liée à celle de Nginx, Naxsi ne fonctionnant que sur le serveur web russe. Pour utiliser Naxsi tout en ayant un autre serveur web, vous devrez installer Nginx+Naxsi en reverse proxy devant le votre.

Sur une distribution Linux de type Debian/Ubuntu, l'installation est packagée : `sudo apt-get install nginx-naxsi`

Il vous faudra alors activer Naxsi dans la configuration de Nginx et générer vos listes blanches à l'aide de l'outil nx_util. Reportez-vous à la documentation en ligne pour cela :

https://code.google.com/p/naxsi/wiki/NxUtil_man

Vous savez maintenant qu'avec les WAF, c'est un nouveau pan de la sécurité web qui s'ouvre à vous : ne passez plus à côté !

Jean-Baptiste Boisseau, Eutech SSII
jb.boisseau@eutech-ssii.com

Guava by Google

2^e partie

Le mois dernier, nous avons fait connaissance avec la bibliothèque Guava. Nous avons découvert comment elle permet d'augmenter la qualité de nos programmes, y compris dans des méthodologies sur lesquelles on ne pensait plus intervenir. Nous avons aussi parlé des objets immutables, pierres angulaires permettant de bien aborder la programmation fonctionnelle.

Dans cette suite, nous allons voir comment installer Guava, car nous avions laissé ce point en suspens. Nous parlerons des fonctions et des prédicats dans le cadre de la programmation fonctionnelle. Nous utiliserons un ensemble de nouvelles collections, ajoutées par la lib et qui deviendront vite indispensables dans vos programmes. En même temps, nous verrons quelques utilitaires pour simplifier vos traitements.

> Installation

Le plus simple, pour ajouter Guava à un projet, est d'en télécharger le JAR (Java Archive) chez Google puis de le placer dans les sources. La plupart des équipes préféreront néanmoins utiliser un logiciel d'automatisation comme Maven, Gradle ou Ivy. Dans le cas de Maven, c'est une simple configuration XML. Ici, on utilise la version « 15.0 » dont la release finale a été publiée durant l'écriture de cet article :

```
<dependency>
  <groupId>com.google.guava</groupId>
  <artifactId>guava</artifactId>
  <version>15.0</version>
</dependency>
```

Pour Gradle, la syntaxe est complètement différente, bien que les informations soient globalement les mêmes :

```
repositories {
  mavenCentral()
}

dependencies {
  compile
    group: 'com.google.guava',
    name: 'guava',
    version: '15.0'
}
```

> Programmation fonctionnelle

Une des premières choses à laquelle on pense lorsqu'on parle de Guava, c'est que la bibliothèque permet d'aborder la programmation fonctionnelle (PF). Bien que les concepts qu'elle propose ne soient pas récents, force est d'admettre qu'ils n'étaient pas à la mode. Mais les choses changent et la PF revient sérieusement en force. La dernière fois, nous avons utilisé le bean « Personne » qui représentait notre domaine d'étude. Pour aller plus loin, voici l'enum « Genre » sur lequel nous allons appliquer des filtres :

```
public enum Genre {
  FEMME(«F»),
  HOMME(«H»);

  private final String code;
```

```
Genre(final String code) {
  this.code = code;
}

public String getCode() {
  return code;
}

public static Genre valueOfByCode(final String code) {
  for (final Genre genre : values()) {
    if (genre.code.equals(code)) {
      return genre;
    }
  }
  throw new IllegalArgumentException(«Code non supporte»);
}
```

Et pour faire bonne mesure, nous allons l'utiliser directement dans la collection « personnes » que nous avons déjà définie :

```
final List<Personne> personnes = Lists.newArrayList(
  new Personne(«Anne», «Dupond», FEMME),
  new Personne(«Bernard», «Lavache», HOMME),
  new Personne(«Céline», «Martini», FEMME),
  new Personne(«Hing», «Chan», HOMME),
  new Personne(«Léon», «Loup», HOMME),
  new Personne(«Thierry», «Lerich», HOMME),
  new Personne(«Vincent», «Duval», HOMME));
```

Le premier objet auquel on doit s'intéresser est le « Predicate » pour lequel il faut redéfinir la méthode « apply ». Celle-ci prend l'objet cible en entrée et renvoie un booléen dont la valeur indique si le prédicat ainsi défini s'applique. Par exemple, on peut écrire « malePredicate » pour identifier un homme :

```
Predicate<Personne> malePredicate = new Predicate<Personne>() {
  @Override
  public boolean apply(final Personne personne) {
    return personne.getGenre() == HOMME;
  }
};
```

Il suffit alors d'utiliser les méthodes statiques « find » et « filter » pour sélectionner respectivement le premier homme d'une liste ou tous les hommes :

```
Personne homme = Iterables.find(personnes, malePredicate);

Iterable<Personne> hommes = Iterables.filter(personnes, malePredicate);
```


Attention : la méthode « filter » lève une exception (NSEE) si elle n'est pas en mesure de trouver un élément qui correspond, comme lorsqu'on recherche un homme dans la liste des femmes. Dans ce cas, on pourra soit employer la méthode « tryFind » soit indiquer une valeur par défaut qui sera retournée si besoin :

```
Personne homme = find(femmes, malePredicate, new Personne("Miss",
"Doubtfire"));
```

Guava offre aussi des prédicats déjà prêts qu'on peut employer directement. Par exemple, pour vérifier que Léon appartient à la fois à la liste des personnes et au set des clients, on pourra composer des prédicats classiques :

```
final Set<Personne> clients = Sets.newHashSet(
    new Personne("John", "Lada"),
    new Personne("Léon", "Loup"),
    new Personne("Vincent", "Zoulot"));

boolean inBothColl = and(in(personnes), in(clients))
    .apply(new Personne("Léon", "Loup"));
```

On notera que les filtres ne s'appliquent pas seulement aux listes et aux sets ; on peut parfaitement filtrer des clés de maps à l'aide de la méthode « filterKeys ».

Pour compléter les prédicats, on va s'intéresser à l'objet « Function » pour lequel on doit également redéfinir la méthode « apply ». Cette fois, celle-ci prend l'objet cible en entrée, mais renvoie un objet d'un autre type cible. Par exemple, pour transformer une personne en une chaîne de caractères (String) avec la concaténation de son prénom et de son nom (trivial), on écrira :

```
Function<Personne, String> nameFunction = new Function<Personne,
String>() {
    @Override
    public String apply(final Personne personne) {
        return personne.getPrenom() + « » + personne.getNom();
    }
};
```

Il suffit ensuite d'utiliser la méthode « transform » pour réaliser la transformation associée :

```
List<String> noms = Lists.transform(personnes, nameFunction);
```

Cette opération s'effectue de façon « lazy », c'est-à-dire à chaque fois qu'on accède à un des éléments (transformés) de la « vue » et sans mise en cache. Si on utilise plusieurs fois un des items, la transformation sera effectuée autant de fois. En revanche, si on n'y accède jamais, aucune opération ne sera réalisée. Cela se fait élément par élément et non pour l'ensemble de la vue. Dans le programme, il faut donc décider si ce comportement « lazy » correspond au besoin ou si on va forcer la transformation une fois pour toutes en injectant la vue dans une nouvelle liste.

“la transformation s'effectue de façon lazy...”

À l'aide de prédicats et de fonctions, on peut donc effectuer des opérations « filter-map » pour filtrer et transformer des collections :

```
List<Personne> hommes = newArrayList(filter(personnes, male
Predicate));
```

```
List<String> noms = transform(hommes, nameFunction);
```

Cette technique oblige toutefois à employer des variables intermédiaires et ne correspond pas complètement à l'objectif. À la place, on va plutôt utiliser les fonctionnalités des « FluentIterables » et les chaîner directement. Pour compléter le tableau, on demandera le résultat sous forme d'immutable :

```
List<String> noms = FluentIterable.from(personnes)
    .filter(malePredicate)
    .transform(nameFunction)
    .toImmutableList();
```

Attention toutefois, bien que Guava incite les développeurs à écrire du code propre, et donc structurellement optimisé, il n'y a rien à espérer du côté des performances au niveau de la FP. Pour cela, il faudra attendre les Lambdas de Java 8.

> Unions, intersections et exclusions de sets ou de maps

Lorsque vous manipulez des sets, vous avez régulièrement besoin d'en établir des unions ou des intersections. Par exemple, Amazon doit déterminer les modes d'envoi possibles pour un panier en se basant sur les modes disponibles pour chaque produit. L'algorithme classique doit justement alterner un calcul d'unions puis d'intersections.

```
Set<String> set1 = ImmutableSet.of("Collissimo", "Kiala",
"Chronopost", "UPS");
Set<String> set2 = ImmutableSet.of("Fedex", "UPS", "Magasin");

Set<String> union = Sets.union(set1, set2);
// à [Collissimo, Kiala, Chronopost, UPS, Fedex, Magasin]
```

Le calcul de l'union n'est pas très difficile en Java classique. Il suffit d'utiliser un set dans lequel on injecte tous les éléments de façon mécanique ; les capacités de déboulonnage des sets font le gros du travail. L'intersection et l'exclusion sont bien plus délicates, surtout sur un nombre plus important de collections à traiter. Encore une fois, avec Guava, cela devient simple :

```
Set<String> inter = Sets.intersection(set1, set2);
Set<String> diff = Sets.difference(set1, set2);
```

Bien entendu, cela ne fonctionne pas uniquement avec des sets. On peut aussi calculer des différences de maps, dont les clés ne sont finalement que des sets :

```
Map<String, Integer> map1 = ImmutableMap.<String, Integer> builder()
    .put("Julie", 25)
    .put("Marc", 34)
    .put("John", 17)
    .build();
Map<String, Integer> map2 = ImmutableMap.<String, Integer> builder()
    .put("Thierry", 45)
    .put("Marc", 34)
    .put("John", 17)
    .put("Arnaud", 13)
    .build();

MapDifference<String, Integer> mapdiff = Maps.difference(map1,
map2);
```

```
// à not equal: only on left={Julie=25}: only on right={Thierry
=45, Arnaud=13}
Map<String, Integer> mapinter = mapdiff.entriesInCommon(); //
à {John=17, Marc=34}
```

> Multiset

En Java, lorsqu'on a besoin d'une collection pouvant recevoir des doublons et dont les éléments sont ordonnés, on utilise une liste. Quand on veut une collection sans doublon et sans ordre significatif, on choisit un set. Et lorsqu'on ne veut pas d'ordre, sans s'interdire les doublons, il n'y a pas de solution. C'est ce manque que comblent les multisets de Guava :

```
Multiset<String> multiset = HashMultiset.create();

multiset.add(«bleu»);
multiset.add(«jaune»);
multiset.add(«vert»);
multiset.add(«jaune»);
multiset.add(«noir»);
multiset.add(«rouge»);
```

Le multiset s'utilise comme un set classique au delta qu'il conserve des informations sur les doublons, ce qu'on voit lorsqu'on imprime les éléments :

```
[noir, rouge, vert, bleu, jaune x 2]
```

Dans cet exemple, on constate que le multiset a bien vu qu'il y avait deux fois l'item « jaune ». Il est également possible de demander le décompte directement à la collection :

```
int nb = multiset.count(«jaune»); // à 2
```

> Multimap

Une multimap est une map de liste, ou plus généralement une map de collection. C'est un concept utilisé par de très nombreux programmes, par exemple pour gérer les achats des clients d'un magasin :

```
Map<Personne, List<String>> map = new HashMap<Personne, List<String>>();
```

En Java « pure », c'est relativement simple à écrire et pas beaucoup plus difficile à manipuler. Il faut toutefois bien gérer quelques pièges comme la nullité éventuelle des valeurs de la map. C'est une tâche rébarbative qu'on s'empresse donc généralement de développer dans une méthode à part :

```
public static void addAchat(Personne client, String produit,
Map<Personne, List<String>> map) {
    List<String> produits = map.get(client);
    if (produits == null) {
        produits = new ArrayList<String>();
        map.put(client, produits);
    }
    produits.add(produit);
}
```

On doit ensuite faire un appel statique, qui n'est ni pratique ni lisible :

```
Personne paul = new Personne(«Paul», «Dupont»);
```

```
addAchat(paul, «tshirt», map);
addAchat(paul, «pull», map);
...
```

En outre, le fait de manipuler une map ne permet pas de savoir en un coup d'œil ce qui se cache derrière. Guava propose donc sa version des multimaps, qui est bien plus simple à définir et à utiliser, et dont le nommage fait passer un message clair aux développeurs :

```
Multimap<Personne, String> multimap = HashMultimap.create();
```

L'instance s'obtient simplement via un « static factory » et ne nécessite pas de préciser les détails d'implémentation. L'utilisation est d'autant plus simple et correspond mieux au modèle objet :

```
multimap.put(paul, «tshirt»);
multimap.put(paul, «pull»);
...
```

> BiMap

Puisqu'on parle de map, continuons sur le sujet. La bibliothèque propose une collection, nommée BiMap, qui est tout simplement une map bijective. Cela signifie qu'une clé est associée à une et une seule valeur et qu'une valeur est associée à une et une seule clé.

On a besoin d'une map bijective lorsqu'on la parcourt à la fois sur la clé et sur la valeur. L'exemple souvent cité est celui d'une boutique dont les employés recherchent un client par son nom ou son numéro de carte de fidélité.

“développer une map bijective est loin d'être trivial...”

Pour programmer une map bijective en Java « pure », il faut utiliser deux maps, avec des clés et des valeurs inversées. Il faut interdire les doublons et, surtout, bien gérer la concurrence et la synchronisation inter-map ; c'est loin d'être trivial, contrairement à ce que propose Guava :

```
BiMap<String, Personne> bimap = HashBiMap.create();

bimap.put(«ABC123», new Personne(«Paul», «Dupont»));
bimap.put(«GHI456», new Personne(«Sophie», «Martini»));
...
Personne sophie = bimap.get(«GHI456»);
```

Une bimap s'utilise comme un map classique. Elle possède la particularité, du fait de la bijection des données, de pouvoir être inversée, c'est-à-dire que les valeurs peuvent servir de clé :

```
BiMap<Personne, String> mapbi = bimap.inverse();
String id = mapbi.get(sophie); // GHI456
```

Et bien entendu, la bimap lève une exception si on essaie d'ajouter une valeur en double :

```
mapbi.put(new Personne(«John», «Do»), «GHI456»); // à IAE value
already present: GHI456
```

> Table

Guava étend le concept de la map en proposant les tables. Alors qu'une map classique fonctionne en deux dimensions, avec ses clés

et ses valeurs, une table travaille sur trois puisqu'elle utilise une double clé :

```
Table<Integer, Genre, Personne> table = HashBasedTable.create();

table.put(21, FEMME, new Personne(«Sophie», «Martini»));
table.put(28, FEMME, new Personne(«Virginie», «Papatis»));
table.put(35, FEMME, new Personne(«Alice», «Bihan»));
table.put(21, HOMME, new Personne(«John», «Labirol»));
table.put(25, HOMME, new Personne(«Julien», «Pong»));
table.put(35, HOMME, new Personne(«Adrien», «Pactola»));
table.put(36, HOMME, new Personne(«Thierry», «Leriche»));
```

Quand on recherche une valeur, on doit donc indiquer deux clés :

```
Personne alice = table.get(35, FEMME);
```

On peut également se contenter d'indiquer une seule clé. Dans ce cas, la recherche renverra une map et non pas une valeur unique. Il faut néanmoins préciser sur quelle clé, désignée en tant que ligne ou colonne, on travaille :

```
Map<Integer, Personne> byColumn = table.column(FEMME); // à Alice,
Sophie, Virginie
Map<Genre, Personne> byLine = table.row(35); // à Alice, Adrien
```

> Range

Depuis quelques versions, Guava nous propose également de travailler avec des intervalles, soit en tant que tels, soit en complément des collections. Un intervalle est défini comme un ensemble de valeurs comprises entre une borne min et une borne max. Les bornes elles-mêmes peuvent appartenir, ou non, à l'intervalle. Dans la suite nous utiliserons des intervalles d'entiers, mais il est aussi possible d'employer des flottants ou des éléments alphanumériques.

“ les bornes peuvent appartenir, ou non, à l'intervalle...”

Lorsque l'intervalle inclut ses bornes, on dit qu'il est fermé. À l'inverse, lorsqu'il les exclut, on dit qu'il est ouvert. Et il est possible de préciser s'il est ouvert ou fermé à gauche comme à droite. On a donc quatre types d'intervalles :

```
Range<Integer> range1 = Range.closed(1, 5); // à [1; 5]
Range<Integer> range2 = Range.open(1, 5); // à ]1; 5[
Range<Integer> range3 = Range.openClosed(1,5); // à ]1; 5]
Range<Integer> range4 = Range.closedOpen(1,5); // à [1; 5[
```

Cette instanciation peut se faire de manière paramétrée, pour la rendre plus dynamique :

```
int min = 3;
int max = 8;
BoundType gauche = CLOSED;
BoundType droite = OPEN;
Range range5 = Range.range(min, gauche, max, droite); // [3; 8[
```

Si vous avez étudié les intervalles à l'école, vous savez déjà qu'on peut définir des intervalles allant d'une valeur jusqu'à l'infini :

```
Range<Integer> infneg = Range.atMost(5); // à ]-∞; 5]
Range<Integer> infpos = Range.atLeast(5); // à [5; +∞[
```

La principale opération que l'on va pouvoir effectuer sur un intervalle est de vérifier qu'il contient une valeur particulière, ce qui peut dépendre des bornes :

```
boolean dedans1 = range1.contains(5); // à true
boolean dedans2 = range2.contains(5); // à false
```

Cette vérification peut se faire aussi sur une liste de valeurs :

```
List<Integer> premiers = Lists.newArrayList(1, 2, 3, 5);
boolean tousDedans = range1.containsAll(premiers); // à true
```

On va surtout pouvoir combiner les intervalles sous forme d'unions et d'intersections :

```
Range<Integer> inter = range1.intersection(range5); // [3; 5]
Range<Integer> union = range1.span(range5); // [1; 7]
```

> Collections avec des Ranges

Quitte à travailler sur des unions, on peut carrément utiliser des sets d'intervalles :

```
RangeSet<Integer> set = TreeRangeSet.create();
set.add(Range.closed(1, 5)); // [1; 5]
set.add(Range.closed(6, 9)); // [6; 9]
set.add(Range.closed(11, 15)); // [11; 15]
```

Et on pourra les utiliser pour vérifier, par exemple, qu'un intervalle est inclus dedans :

```
boolean dedans = set.encloses(Range.closed(4, 5));
```

Les intervalles peuvent aussi s'utiliser avec des maps en tant que clé :

```
RangeMap<Integer, Personne> map = TreeRangeMap.create();

map.put(closed(1, 5), new Personne(«Julie», «Adobi»));
map.put(closed(10, 15), new Personne(«Marc», «Dubois»));
map.put(closed(20, 25), new Personne(«Luc», «Ciel»));
...
```

Quand on voudra une valeur, la map ira la chercher dans les intervalles. Cela fonctionne finalement comme si la map contenait de nombreuses clés. Ceci en sachant qu'il faudra garder en mémoire que ça n'utilise pas seulement des entiers, c'est-à-dire que ça n'utilise pas que des valeurs discrètes :

```
Personne marc = map.get(12);
```

> Conclusion

Cet article traite principalement des collections, peut-être un peu trop. J'espère que vous n'en faites pas une overdose car nous n'avons fait qu'effleurer la surface de ce que propose Guava dans ce domaine. Nous avons découvert un jeu de collections additionnelles bien pratiques au quotidien. Nous avons également discuté de programmation fonctionnelle. Sur ce dernier point, il est important de noter, comme le précisent les développeurs de l'équipe Guava, qu'il convient de ne pas en abuser. En effet, une simple boucle « for » est souvent plus efficace et plus lisible que l'emploi d'une classe anonyme.

Guava : <https://code.google.com/p/guava-libraries/>



Thierry Leriche-Dessirier
Architecte JEE freelance
<http://www.icauda.com>

WebAPI : créer son API pour manipuler facilement http

Le protocole http ne doit pas être vu uniquement comme permettant de servir des pages HTML pour construire un site web. Aujourd'hui, nous utilisons http dans toutes les applications ou presque : que ce soit pour construire des services web SOAP ou REST, pour échanger des données, voire même pour faire de la communication temps réel (avec SignalR par exemple).

Dans cet article nous allons découvrir la réponse que propose Microsoft à ce besoin grandissant pour un développeur web de manipuler le protocole http. Nous ne nous intéresserons pas ici à la création de service web utilisant la grammaire SOAP, car c'est plus le domaine du framework WCF, mais plutôt au développement de services web RESTful qui deviennent de plus en plus nombreux sur le web. Puis, nous explorerons les aspects relatifs à la sécurité dans le cadre de l'exposition d'une API REST. Enfin nous verrons comment publier des données via http en utilisant le protocole OData.

> Architecture REST : l'exemple des API Twitter et Facebook

Mais avant de rentrer dans l'implémentation proprement dite, regardons rapidement ce que proposent les géants du web de nos jours. Twitter, Facebook, ainsi que de nombreux autres, comme eBay ou Google, fournissent désormais systématiquement un modèle de programmation http autour des fonctionnalités qu'ils proposent.

Il est par exemple possible, via une API REST dédiée et documentée, de poster des messages sur un mur Facebook, ou de consulter des tweets. Pour permettre au plus grand nombre de clients de consommer ces APIs, quoi de plus normal que de choisir le protocole le plus célèbre du web : le protocole http.

Même si vous ne construisez pas des applications aussi ambitieuses que Facebook ou Twitter, il est de plus en plus fréquent de proposer en sus de votre application, une API permettant de la « programmer » pour permettre à d'autres de l'étendre ou de l'utiliser dans des

contextes que vous n'auriez pas forcément imaginés.

L'utilisation de la WebAPI de Microsoft vous permettra de construire rapidement et de manière standardisée une API consommable en http qui exposera vos services très simplement à tous.

> Implémenter la WebAPI Package NuGet

Pour implémenter une API bâtie avec la WebAPI, il ne vous faut qu'un ensemble de packages NuGet pour commencer ! Une application console peut même faire l'affaire, car ces packages NuGet contiennent tout ce qu'il faut pour émuler un serveur capable de répondre à des requêtes http.

Néanmoins, le plus simple est tout de même de partir d'un site web ASP.NET MVC. Dans les derniers templates disponibles dans Visual Studio, on peut constater la présence d'un modèle de site ASP.NET MVC avec WebAPI, comme le montre l'image ci-dessous : [Fig.1].

Notez que ce template de projet propose déjà tout un tas de choses précablées pour héberger une API via la WebAPI. Si comme moi vous souhaitez avoir plus de contrôle sur ce que vous allez réellement utiliser, il est possible de partir du template de projet vide qui contient également les packages NuGet pour utiliser la WebAPI.

Nous verrons plus tard comment héberger votre API dans une application console, en utilisant la technologie OWIN (Open Web Interface for .Net). Mais pour l'instant nous débuterons avec un hébergement ASP.NET MVC car cela est le plus simple à mettre en œuvre.

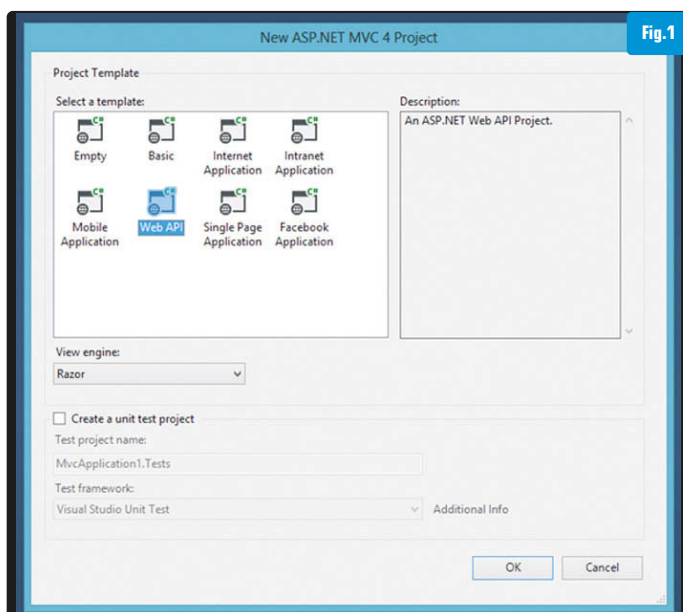
Votre première API web

Maintenant que votre site MVC est prêt, nous allons pouvoir créer notre première API. Nous allons implémenter une méthode de recherche d'entités à la manière de la recherche via l'API Facebook. Avec cette API, pour voir le détail de la société ITELIOS, voici la requête http qu'il faut émettre <https://graph.facebook.com/itelios> et voici un extrait du résultat JSON que l'on obtient : [Fig.2]. Si nous souhaitons implémenter la même requête via notre site web MVC en utilisant la WebAPI, il nous faut alors ajouter un contrôleur WebAPI avec le code ci-dessous : [Fig.3]. En héritant de la classe **ApiController** notre contrôleur sera en mesure de répondre aux requêtes http qui lui arriveront via la route configurée plus tôt dans notre projet MVC : [Fig.4]. Si on exécute la requête http GET « /api/facebook/itelios » nous recevrons en réponse le flux XML suivant : [Fig.5].

Convention over configuration

Mais comment se fait-il que notre requête http sache automatiquement qu'il faille appeler la méthode **GetFacebookObject()** ?

Tout simplement car la WebAPI fonctionne dans un mode dit



« convention over configuration » que l'on pourrait traduire par « les conventions d'abord, la configuration ensuite » ; ceci signifie que par défaut, un certain nombre de conventions sont mises en places lors de l'utilisation de cette API.

L'une de ces conventions dit qu'une méthode commençant par le préfixe **Get** dans un contrôleur WebAPI est une méthode qui répondra à la requête http GET sur la route par défaut. Ainsi, l'appel de notre URL déclenche bien l'exécution de la méthode **GetFacebookObject** qui aurait pu tout aussi bien s'appeler **GetFacebookEntity** par exemple.

De même, les méthodes commençant par les préfixes **Put**, **Post** ou **Delete** seront associées aux verbes http correspondants. Cela permet donc d'écrire facilement une API compatible avec l'architecture REST qui fait grand usage des verbes http pour décrire la manipulation de ressources via http.

Négociation de contenu

Nous avons vu précédemment que la réponse de notre API fournit des données au format XML. Qu'en est-il si nous souhaitons obtenir ces données au format JSON, à l'image de l'API Facebook prise en exemple ?

Et bien cela est possible par une autre convention de la WebAPI qui concerne la négociation de contenu. Par défaut, l'API renverra des résultats au format XML, mais si nous souhaitons obtenir ces

mêmes résultats sous format JSON, il est possible de le faire sans modifier une ligne de code.

Il suffit simplement d'indiquer dans notre requête http à notre API que nous souhaitons obtenir un document JSON. Pour cela, il faut juste ajouter à notre requête http un en-tête « **Accept : text/json** » qui indiquera à la WebAPI de sérialiser notre objet de retour au format JSON plutôt que XML.

Notez qu'il est possible d'étendre ce comportement par défaut en implémentant l'interface **IContentNegotiator**, ce qui permet, si on le souhaite, de gérer des sérialisations dans des types particuliers.

Implémenter la modification d'entité

Admettons désormais que l'on souhaite que notre API puisse modifier une entité par le biais d'une requête http PUT. Avec ce que l'on vient de voir, il suffirait alors de rajouter une méthode comme celle qui suit à notre contrôleur : [Fig.6].

Notons plusieurs changements ici par rapport à la méthode **GetFacebookObject**. Une API REST doit « par convention » renvoyer un code d'erreur http 404 en cas d'objet non trouvé lors d'une modification. De même, si l'objet existe bien et qu'il a pu être modifié, un code http 204 (no-content) doit être renvoyé au client.

C'est pourquoi notre méthode d'enregistrement **PutFacebookObject** renvoie une instance de la classe **HttpResponseMessage** qui permet de formater la réponse http servie au client qui consomme cette API.

Voilà comment implémenter avec très peu de code une API compatible avec les standards de l'architecture REST en se basant sur la WebAPI. Bien sûr, il est possible de surcharger les conventions par défaut pour les verbes http afin d'écrire une API plus appropriée à vos besoins, mais dans la majorité des cas les conventions par défaut nous suffisent.

Présenter la documentation de votre API

De nos jours, les APIs REST sont de plus en plus fréquentes. Mais au contraire d'un service web classique utilisant le langage WSDL pour décrire les opérations disponibles sur un service, il n'est pas possible de faire de même avec une API REST. Ainsi, les utilisateurs de votre API s'attendent généralement à retrouver une documentation en ligne de votre API leur permettant de savoir : comment elle

Fig.2

```

{
  about: "Société innovante spécialisée dans la conception, l'intégration et la mise en œuvre sites internet e",
  category: "Internet/software",
  category_list: [
    {
      id: "2256",
      name: "Internet/Software"
    }
  ],
  description: "Société innovante spécialisée dans : la conception, l'intégration et la mise en œuvre de plat",
  expertise: "l'expertise des Technologies Microsoft - les stratégies e-business et e-marketing",
  is_published: true,
  location: {
    street: "10 rue de Sèze",
    city: "Paris",
    state: "",
    country: "France",
    zip: "75009",
    latitude: 48.870331042101,
    longitude: 2.3270276902591
  },
  phone: "01 40 98 02 00",
  talking_about_count: 14,
  username: "Itelios",
  website: "http://twitter.com/Itelios",
  were_here_count: 17,
  id: "399742340044375",
  name: "Itelios",
  link: "https://www.facebook.com/Itelios",
  likes: 85,
  cover: {
    cover_id: 454239541261321,
    source: "https://photos-h.ak.fcdn.net/hphotos-ak-snb3/e720x720/552505_454239541261321_889748022_n.jpg",
    offset_y: 0,
    offset_x: 0
  }
}

```

Fig.3

```

1 using System;
2 using System.Net;
3 using System.Net.Http;
4 using System.Web.Http;
5 using Programmez.WebAPI.Models;
6
7 namespace Programmez.WebAPI.Controllers
8 {
9     /// <summary>Permet une émulation de l'API Facebook via la WebAPI</summary>
10    public class FacebookController : ApiController
11    {
12        /// <summary>Renvoie un objet Facebook</summary>
13        /// <param name="id">Identifiant de l'objet</param>
14        /// <returns>Structure représentant l'objet Facebook demandé</returns>
15        public FacebookObject GetFacebookObject(string id)
16        {
17            if (String.Equals(id, "itelios", StringComparison.InvariantCultureIgnoreCase))
18            {
19                return new FacebookObject()
20                {
21                    Description = "Société innovante spécialisée dans...",
22                    Username = "Itelios",
23                    Website = "http://twitter.com/Itelios"
24                };
25            }
26            else return null;
27        }
28    }
29 }
30

```

Fig.4

```

1 using System;
2 using System.Web.Http;
3 using Programmez.WebAPI.Handlers;
4
5 namespace Programmez.WebAPI
6 {
7     public class WebAPIConfig
8     {
9         public static void Register(HttpConfiguration config)
10         {
11             config.Routes.MapHttpRoute("DefaultApi", "api/{controller}/{id}", new { id = RouteParameter.Optional });
12         }
13     }
14 }

```

Fig.5

```

*FacebookObject xmlns="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsi="http://schemas.datacontract.org/2004/07/Programmez.WebAPI.Models"
  <Description>Société innovante spécialisée dans...</Description>
  <Username>Itelios</Username>
  <Website>http://twitter.com/Itelios</Website>
  <FacebookID>399742340044375</FacebookID>

```

Fig.6

```

/// <summary>Mise à jour d'un objet Facebook existant</summary>
/// <param name="id">Identifiant de l'objet</param>
/// <param name="data">Structure représentant l'objet Facebook devant être modifié</param>
/// <returns>Code HTTP 204 si l'enregistrement est OK. Code HTTP 404 si l'objet n'existe pas</returns>
public HttpResponseMessage PutFacebookObject(string id, FacebookObject data)
{
    if (String.Equals(id, data.Username))
    {
        // TODO : faire la modification de l'objet
    }
    return Request.CreateResponse(HttpStatusCode.NoContent);
}
else return Request.CreateResponse(HttpStatusCode.NotFound);
}

```

fonctionne, quelles sont les paramètres attendus, les valeurs de retours des appels http, etc...

Bien que très importante, la documentation technique fait pourtant partie des choses qu'un développeur n'aime pas spécialement mettre en place. De plus, en cas de changement dans l'API il est facile d'oublier qu'il faut « mettre à jour le manuel ».

Du coup, et bien qu'il soit possible d'écrire soi-même sa propre documentation en utilisant les fonctionnalités de production HTML du site MVC hébergeant notre API, un outil permettant de le faire de manière automatique serait le bienvenu. Encore plus, si cet outil s'appuyait sur une documentation déjà présente dans notre code : les commentaires XML de méthodes.

Fort heureusement cet outil existe sous forme de package NuGet. Pour l'installer il suffit d'exécuter la ligne de commande suivante :

```
Install-Package Microsoft.AspNet.WebApi.HelpPage -Pre
```

Cela aura pour effet d'ajouter une **area MVC** nommée **help** dans votre projet. Cette **area** contiendra tout ce qu'il faut pour présenter une documentation en ligne automatique de votre API. Par défaut, elle ne s'appuie pas sur les commentaires XML, mais il est possible de lui indiquer de le faire, en activant une ligne de code dans un des fichiers fournis avec le package NuGet : [Fig.7]. Ne reste plus alors qu'à naviguer jusqu'à cette **area** pour se rendre compte que tous les contrôleurs décorés par des commentaires XML sont désormais disponibles dans cette page d'aide. Ceci avec moult détails expliquant le fonctionnement de chaque action et exemple de sérialisation XML/JSON des paramètres des méthodes [Fig.8].

Ce package NuGet est présent par défaut dans un projet ASP.NET MVC 4.0 de type WebAPI. S'il ne vous convient pas directement, il est possible d'en télécharger d'autres dans les repository NuGet qui fonctionnent de manière à peu près similaire; dans la majorité des cas, celui-ci fait l'affaire.

Héberger votre API dans une application console

Jusqu'à présent nous avons vu comment héberger notre API au sein d'un projet MVC en utilisant la WebAPI. Mais celle-ci ne nécessite pas forcément un **host MVC** ni même WebForms pour fonctionner. Il est possible de faire du **self-hosting** dans une application quelconque pour peu qu'on ajoute les bons packages NuGet et qu'on utilise la technologie OWIN (Open Web Interface for .Net) développée par Microsoft.

Pour cela, et en prenant le cas le plus basique de l'application console, il nous suffit donc d'ajouter le package NuGet via la ligne de commande suivante :

```
/// <summary>
/// Use this class to customize the Help Page.
/// For example you can set a custom <code>IHttpDescriptionProvider</code> to supply the documentation
/// or you can provide the samples for the requests/responses.
/// </summary>
public static class HelpPageConfig
{
    public static void Register(HttpConfiguration config)
    {
        // Uncomment the following to use the documentation from XML documentation file.
        config.SetDocumentationProvider(new XmlDocumentationProvider(HttpContext.Current.Server.MapPath("~/App_Data/XmlDocument.xml")));
    }
}
```

Fig.7

ASP.NET Web API Help Page

Introduction

Provide a general description of your APIs here.

Facebook

API	Description
GET api/Facebook?id	Renvoie un objet Facebook
PUT api/Facebook?id	Mise à jour d'un objet Facebook existant

Fig.8

```
Install-Package Microsoft.AspNet.WebApi.OwinSelfHost -Pre
```

Une fois ce package installé avec toute ses dépendances, il va nous falloir créer une classe permettant d'initialiser notre serveur http capable de répondre à des requêtes sur notre API. Cela peut-être n'importe quelle classe de votre projet du moment qu'elle implémente une méthode avec la signature **void Configuration(IAppBuilder)**. Cette méthode sera le point d'entrée de votre configuration pour la WebAPI. Elle devra contenir un code similaire au code ci-dessous : [Fig.9].

Il ne reste alors plus qu'à créer les contrôleurs de notre choix, héritant de la classe **ApiController** pour avoir une API hébergée dans notre application console. Ces contrôleurs se comportent exactement comme nous l'avons vu précédemment avec les mêmes « convention over configuration ».

Enfin, pour démarrer notre serveur OWIN **self-hosted**, il faudra ajouter le code suivant dans la méthode principale de notre application console : [Fig.10].

Et voilà, par l'intermédiaire d'OWIN vous êtes désormais capable de servir des requêtes et réponses http via une application console. Bien sûr, IIS s'avère être un moyen plus efficace de procéder, mais il arrive parfois qu'on ne puisse pas l'utiliser, et dans ce cas le **self-hosting** est une alternative tout à fait valable.

> Sécuriser vos API construites avec WebAPI

Nous allons maintenant voir comment répondre à une problématique fréquente lorsqu'on expose une API publiquement sur http : la gestion de l'authentification et de la sécurité. En effet, peut-être qu'à l'instar de Facebook ou Twitter, nous souhaitons limiter les requêtes à notre API aux personnes autorisées.

Pour cela, Facebook et Twitter s'appuient sur un protocole d'authentification nommé **OAuth**. Bien que très pratique car très documenté et très utilisé, et bien qu'étant supporté par le framework .NET, nous allons nous restreindre à une version plus simple dans le cadre de notre exemple. Supposons que l'on souhaite que les requêtes http ne soient autorisées que si l'en-tête de celles-ci contient une clé secrète dite **API Secret Key**. Nous pourrions imaginer que le code de chacune de nos méthodes irait vérifier la présence de cette clé secrète dans les en-têtes http via la collection exposée par **HttpContext.Current.Request.Headers**. Mais cela serait fastidieux et répétitif, voyons plutôt comment inspecter l'ensemble des requêtes http émises vers notre API.

Création d'un gestionnaire de messages

La WebAPI nous propose une classe permettant de réaliser cette inspection de requêtes http par le biais de la classe **DelegatingHandler**. Via cette classe nous sommes alors en mesure d'écrire un gestionnaire de messages http qui va recevoir la requête http en

```
public class WebApiInitializer
{
    public void Configuration(IAppBuilder appBuilder)
    {
        HttpConfiguration config = new HttpConfiguration();
        config.Routes.MapHttpRoute("DefaultApi", "api/{controller}/{id}", new { id = RouteParameter.Optional });
        appBuilder.UseWebApi(config);
    }
}
```

Fig.9

```
public class Program
{
    public static void Main(string[] args)
    {
        using (WebApp.Start<WebApiInitializer>("http://webapi.demo.local:15579/"))
        {
            Console.ReadLine();
        }
    }
}
```

Fig.10

paramètre et effectuer dessus tout traitement nécessaire, comme par exemple : vérifier la présence d'un en-tête particulier.

Il suffit pour cela d'implémenter la méthode **SendAsync** de la classe de base. Cette méthode est asynchrone et renverra donc un objet de type **Task<T>** avec, dans notre cas «T» qui sera de type **HttpResponseMessage** correspondant à la réponse http de retour. Voici ci-dessous une implémentation vérifiant la présence d'un en-tête **APIKey** dans les headers http de la requête : [Fig.11].

Branchez le gestionnaire sur les requêtes http

Il reste maintenant à brancher ce gestionnaire de messages sur les requêtes http que l'on souhaite inspecter. Différentes options se présentent à nous, car nous pouvons vouloir attacher ce gestionnaire sur :

- Toutes les requêtes http arrivant sur la plateforme,
- Seulement les requêtes http émises sur une route particulière

Dans le premier cas, le gestionnaire s'enregistre au niveau de la configuration WebAPI globale comme suit : [Fig.12].

Dans le second cas, au moment de la création de la route, il faut ajouter ce gestionnaire de message comme suit : [Fig.13].

Notez que si vous configurez plus d'un gestionnaire de messages pour une requête http donnée, ceux-ci s'exécuteront dans l'ordre dans lequel ils sont enregistrés. En cas d'erreur dans l'un de ces gestionnaires, ou de sortie de réponse http précoce, les autres gestionnaires dans la chaîne ne seront pas exécutés. Il est donc préférable de mettre ces gestionnaires de messages en tête de liste lorsqu'il s'agit de vérifier une quelconque authentification.

> Exposer des données via le protocole ODATA

Présentation

OData (ou Open Data Protocol, <http://www.odata.org>) est un protocole développé par Microsoft et permettant d'accéder à des jeux de données via http.

Bien qu'utilisable dans la gamme des produits de l'éditeur, notamment Excel, avec l'extension PowerPivot, ce protocole est également implémenté au travers de nombreuses librairies client (pour consommer une source OData) ou serveur (pour produire une source OData).

Côté client, il existe des versions pour Android et iOS, mais également des connecteurs pour Joomla ou Drupal, ou bien encore JavaScript. Côté serveur, les applications Java ou PHP disposent de modules OData, tout comme les serveurs NodeJS et même les serveurs de bases de données (SQL Server bien entendu, mais également MySQL).

Le but de Microsoft est donc d'imposer ce protocole comme un standard de l'échange de données relationnelles par le biais de requêtes http. En ce sens, il est assez similaire aux protocoles ODBC et JDBC mais dans un contexte web.

D'un point de vue plus technique, l'architecture du protocole OData repose sur l'exposition de fonctions CRUD (Create / Read / Update / Delete) via une API REST. Le format des données échangées est du XML (ATOM Publishing) ou du JSON. Enfin, OData supporte des requêtes de métadonnées permettant aux clients de « découvrir » la structure des données exposées par la source OData.

OData peut donc être vu comme une API REST fonctionnant ainsi à merveille avec la WebAPI.

Package NuGet

Pour nous éviter de devoir ré-implémenter le protocole ODATA systématiquement dans notre code, Microsoft a mis à notre disposition des extensions nous permettant d'exposer des données dans ce format en passant par la WebAPI.

Pour cela, il nous faut tout d'abord ajouter un package NuGet à notre site web. Ce package s'installe avec la ligne de commande suivante :

```
Install-Package Microsoft.AspNet.WebApi.OData -Pre
```

Une fois le package installé, il est temps d'exposer le point d'entrée http pour permettre l'accès à nos données. Ce point d'entrée sera une URL décrivant les données disponibles dans la source OData. Voici un extrait de code montrant comment l'enregistrer dans un site web ASP.NET MVC : [Fig.14].

A partir d'une instance de la classe **ODataConventionModelBuilder** on peut ensuite créer une route MVC avec le template d'URL « /odata/ », qui exposera les données de notre modèle. Si on consulte cette URL dans un navigateur, voici la réponse que l'on obtient : [Fig.15].

Notre source de données OData est vide, il faut maintenant choisir les données que l'on veut exposer via http.

Définir les données que l'on souhaite exposer

Pour cela, nous allons créer un modèle **Entity Framework** basé sur la base de données Microsoft Adventure Works. Pour simplifier notre exemple, nous ne prendrons en compte que la table Employee. Une fois ce modèle d'entités disponible, nous pouvons alors reprendre la méthode **Register** pour lui demander d'exposer l'entité Employee générée pour nous par **Entity Framework**.

Voici les lignes de code que l'on doit ajouter : [Fig.16].

```
public class ApiSecretKeyHandler : DelegatingHandler
{
    private const string SecretKey = "WebApiSecretKey";

    protected override Task<HttpResponseMessage> SendAsync(HttpRequestMessage request, CancellationToken cancellationToken)
    {
        bool authorized = false;

        if (request.Headers.Contains("APIKey"))
        {
            string apiKey = request.Headers.GetValues("APIKey").LastOrDefault();
            authorized = apiKey == ApiSecretKeyHandler.SecretKey;
        }

        if (authorized) return base.SendAsync(request, cancellationToken);
        else
        {
            TaskCompletionSource<HttpResponseMessage> forbiddenResponse = new TaskCompletionSource<HttpResponseMessage>();
            forbiddenResponse.SetResult(new HttpResponseMessage(HttpStatusCode.Forbidden));
            return forbiddenResponse.Task;
        }
    }
}
```

Fig.11

```
public class HandlerConfig
{
    public static void Register(HttpConfiguration config)
    {
        config.MessageHandlers.Add(new CustomHttpHeaderHandler());
    }
}
```

Fig.12

```
public class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        config.Routes.MapHttpRoute("DefaultApi", "api/{controller}/{id}", new { id = RouteParameter.Optional }, null, new ApiSecretKeyHandler());
    }
}
```

Fig.13

```
public class ODataConfig
{
    public static void Register(HttpConfiguration config)
    {
        ODataConventionModelBuilder modelBuilder = new ODataConventionModelBuilder();
        config.Routes.MapODataRoute("ODATA", "odata", modelBuilder.GetEdmModel());
    }
}
```

Fig.14

```
<?xml version="1.0" encoding="utf-8" ?>
<workspace>
  <odata>
    <entity type="text">Default</entity>
  </workspace>
</?xml>
```

Fig.15

La seconde ligne de la méthode permet d'enregistrer l'entité dans le modèle OData. La suivante, permet de configurer la clé primaire qui sera utilisée par OData lorsque ces données sont exposées en http. Du coup, en rafraîchissant notre URL consultée précédemment, voici la nouvelle réponse : [Fig.17].

Les données sont exposées en XML en suivant la « grammaire » **ATOM Publishing**. Ce flux XML indique que la source de données OData expose une collection nommée Employees et que celle-ci est consultable en suivant le lien « /odata/Employees » (attention la casse est importante ici).

Cependant, si on consulte cette URL, nous obtenons pour le moment une réponse http 404. En effet, il faut désormais mettre en place dans notre site MVC un contrôleur qui répondra aux demandes http arrivant sur cette URL.

Création du contrôleur OData

Bien que l'on puisse utiliser un contrôleur WebAPI classique pour répondre aux requêtes sur cette route OData, il est plus pratique d'utiliser un contrôleur OData qui prend en charge pour nous un grand nombre des opérations liées à ce protocole.

Nous allons donc créer un contrôleur héritant de la classe **EntitySetController<TEntity, TKey>**. Le type générique TEntity représente le type de l'entité que l'on souhaite exposer. Et le type générique TKey indique le type de la clé primaire pour cette entité.

Le nom que portera ce contrôleur sera celui de la route OData que l'on souhaite rendre disponible; dans notre cas, il s'agit d'« Employees ». Voici donc le code de base de ce contrôleur : [Fig.18]. Notez que le constructeur du contrôleur initialise le contexte **Entity Framework**. De même, nous surchargeons la méthode **Dispose** pour permettre de libérer correctement ce contexte.

Il reste maintenant à ajouter une action permettant de répondre à la route que nous souhaitons mettre en place. Cette route correspond à une requête http GET sur ce contrôleur. La méthode à surcharger est donc la méthode **Get()** de la classe de base **EntitySetController<TEntity, TKey>** : [Fig.19].

Cette méthode renvoie directement les entités de type **Employee** associé à notre contexte **Entity Framework**. Cependant, comme toujours avec l'interface **IQueryable<T>**, la requête ne sera exécutée que lorsque le résultat de la requête sera énuméré. Cela permet donc d'implémenter le protocole OData, et, notamment, les fonctions de pagination, de sélection et de tri client-side.

Notons également que pour permettre ce comportement, il faut décorer la méthode avec l'attribut **QueryableAttribute**. Cet attribut propose une configuration des opérations possibles par le client sur cette route. Ici nous avons choisi de limiter la taille des jeux de résultats à 10 enregistrements maximum.

Si on consulte désormais l'URL correspondant à cette route (« /odata/Employees »), cela nous renvoie un résultat de la forme

suivante : [Fig.20].

Notons que les attributs de sélection du protocole OData sont supportés directement par le biais de l'action **Get()**. Ainsi, si on souhaite rechercher le top 5 de tous les employés nés avant 1950, voici la requête http correspondante : /odata/Employees?\$filter=year(BirthDate)%20le%201950&\$top=5. Et voici le résultat renvoyé par cette requête http : [Fig.21].

Puisque nous utilisons **Entity Framework** pour accéder aux données et que nous renvoyons une interface **IQueryable<T>** le filtre précédent s'applique donc bien côté SQL grâce à LINQ-To-Entities.

Support des opérations de mise à jour

OData permet de requêter des données (le R de CRUD), mais égale-

```

1 using System;
2 using System.Linq;
3 using System.Net;
4 using System.Net.Http;
5 using System.Web.Http;
6 using System.Web.Http.OData;
7 using Programmez.WebAPI.Models;
8
9 namespace Programmez.WebAPI.Controllers
10 {
11     public class EmployeesController : EntitySetController<Employee, int>
12     {
13         private AWEntities DBContext { get; set; }
14
15         public EmployeesController()
16         {
17             this.DBContext = new AWEntities(false);
18         }
19
20         protected override void Dispose(bool disposing)
21         {
22             this.DBContext.Dispose();
23             base.Dispose(disposing);
24         }
25     }
26 }

```

Fig.18

```

[Queryable(PageSize = 10)]
public override IQueryable<Employee> Get()
{
    return this.DBContext.Employee.AsQueryable();
}

```

Fig.19

```

{
  odata.metadata: "http://webapi.demo.local/odata/$metadata#Employees",
  - value: [
    - {
      BusinessEntityID: 1,
      NationalIDNumber: "295847284",
      LoginID: "adventure-works\ken0",
      OrganizationLevel: 0,
      JobTitle: "Chief Executive Officer",
      BirthDate: "1963-03-02T00:00:00",
      MaritalStatus: "S",
      Gender: "M",
      HireDate: "2003-02-15T00:00:00",
      SalariedFlag: true,
      VacationHours: 99,
      SickLeaveHours: 69,
      CurrentFlag: true,
      rowguid: "f01251e5-96a3-448d-981e-0f99d789110d",
      ModifiedDate: "2008-07-31T00:00:00"
    },
    + {...},
    + {...},
    + {...},
    + {...},
    + {...},
    + {...},
    + {...},
    + {...},
    + {...}
  ],
  odata.nextLink: "http://webapi.demo.local/odata/Employees?$Skip=10"
}

```

Fig.20

```

public class ODataConfig
{
    public static void Register(HttpConfiguration config)
    {
        ODataConventionModelBuilder modelBuilder = new ODataConventionModelBuilder();
        EntitySetConfiguration<Employee> employees = modelBuilder.EntitySet<Employee>("Employees");
        employees.EntityType.HasKey(e => e.BusinessEntityID);

        config.Routes.MapODataRoute("ODATA", "odata", modelBuilder.GetEdmModel());
    }
}

```

Fig.16

```

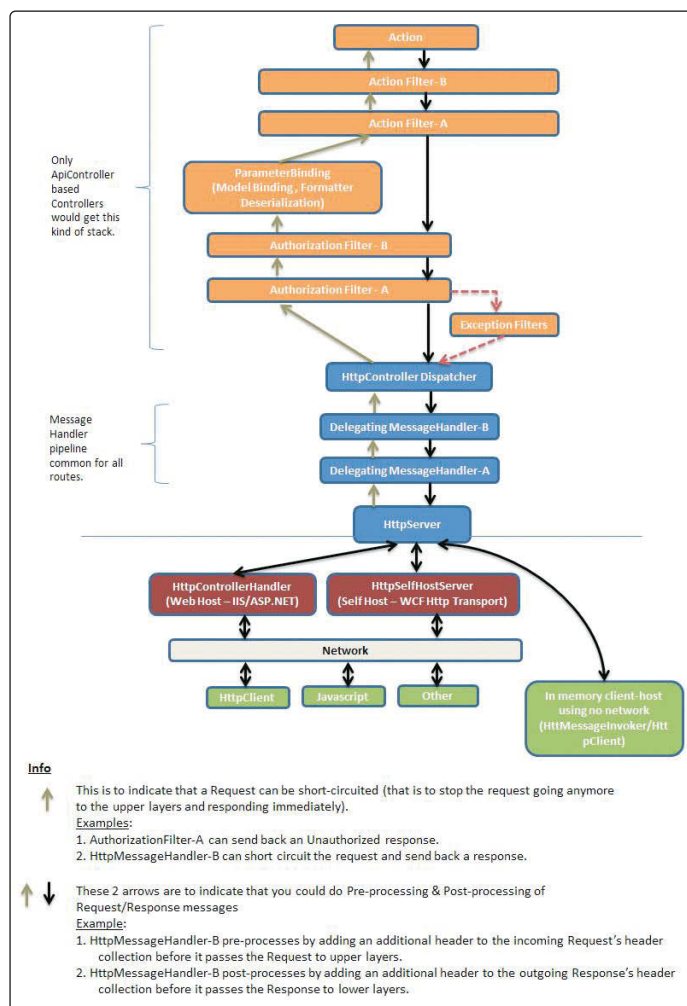
<?xml version="1.0" encoding="utf-8" ?>
<service xmlns="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom" xml:base="http://webapi.demo.local/odata/">
  <ovozKopace>
    <atom:title type="text">Default</atom:title>
    <collection href="Employees">
      <atom:title type="text">Employees</atom:title>
    </collection>
  </ovozKopace>
</service>

```

Fig.17

ment de gérer les opérations de création / mise à jour / suppression. Si on prend l'exemple de la méthode de mise à jour, on peut alors ajouter au contrôleur la méthode **Patch** pour supporter la mise à jour partielle de données : [Fig.22].

Ce code permet de récupérer l'entité **Employee** originelle par son identifiant, puis d'appliquer la différence envoyée par le biais de la requête http PATCH et enfin, de sauvegarder les mises à jour en base de données. D'autres méthodes sont disponibles pour être surchargées sur le contrôleur héritant de la classe **EntitySetController<TEntity, TKey>**, citons par exemple **Delete()**, **Put()**, **Post()**, permettant de compléter les opérations CRUD vues jusqu'à présent.



Pile webapi de ASP.Net MVC4

> Conclusion

Nous avons vu tout au long de cet article, que la WebAPI permet de manipuler très facilement le protocole http pour l'adapter aux besoins du web d'aujourd'hui. Et nous sommes loin d'avoir tout couvert, car il reste énormément de sujets adressables via ce modèle de programmation.

Cette API est en constante évolution et, via les nombreux packages NuGet, peut s'intégrer très rapidement dans n'importe lequel de vos projets .NET en seulement quelques clics.

Pour rester à jour ou en découvrir davantage, n'hésitez pas à consulter le site officiel de Microsoft via l'URL <http://www.asp.net/web-api> qui présente de nombreux tutoriaux sur les grandes fonctionnalités présentées ici.



Guillaume Beaumont
Architecte Senior - Itelios

itelios

```

{
  odata.metadata: "http://webapi.demo.local/odata/$metadata#Employees",
  - value: [
    - {
      BusinessEntityID: 5,
      NationalIDNumber: "695256908",
      LoginID: "adventure-works\gail0",
      OrganizationLevel: 3,
      JobTitle: "Design Engineer",
      BirthDate: "1946-10-29T00:00:00",
      MaritalStatus: "M",
      Gender: "F",
      HireDate: "2002-02-06T00:00:00",
      SalariedFlag: true,
      VacationHours: 5,
      SickLeaveHours: 22,
      CurrentFlag: true,
      rowguid: "ec84ae09-f9b8-4a15-b4a9-6ccbab919b08",
      ModifiedDate: "2008-07-31T00:00:00"
    },
    + {...},
    + {...},
    + {...},
    + {...}
  ]
}

```

Fig.21

```

public override HttpResponseMessage Patch(int key, Delta<Employee> patch)
{
  Employee original = this.DBContext.Employee.Find(key);
  patch.Patch(original);

  this.DBContext.SaveChanges();

  return Request.CreateResponse(HttpStatusCode.NoContent);
}

```

Fig.22



100%
cloud
computing

Actualités, avis d'experts, analyses, stratégies, programmation...

Je démarre la programmation **Android**...



La guerre des smartphones fait rage et chaque constructeur doit choisir un système d'exploitation. Un grand choix s'impose à eux avec les cadors du marché qui sont : Android développé par Google, iOS par Apple et Windows Phone par Microsoft. On trouve également d'autres systèmes comme Blackberry OS ou encore les nouveaux Ubuntu Phone, Firefox OS, Tizen, etc. Suite à l'accroissement continu d'Android sur les smartphones, de nombreux programmeurs s'intéressent au développement d'applications sur ce système d'exploitation. A l'ETNA nous nous sommes intéressés à l'éventualité de créer une application Android pour nos étudiants. Commençons par une petite présentation d'Android.

ANDROID C'EST QUOI ?

Android est un système d'exploitation open source développé par une petite startup rachetée par Google. Il a été lancé en 2007 et a vite pris une place importante sur le marché des smartphones. Actuellement Android représente plus de 75 % du marché des systèmes d'exploitation sur smartphone. Son service de téléchargement d'applications, appelé Play Store, a surpassé récemment l'APP Store d'Apple en nombre de téléchargements.

LANGAGE DE DÉVELOPPEMENT

> JAVA

Développer sous Android n'est pas très compliqué, il suffit de bien connaître le JAVA et d'être curieux. Les programmes destinés à être utilisés sous Android sont écrits en JAVA. Cependant, il faut savoir qu'un programme fonctionnant sur la machine virtuelle JAVA (JVM) ne fonctionnera pas sous Android. En effet, Android inclut une machine virtuelle nommé Dalvik qui permet d'exécuter le même programme prévu pour la plate-forme JAVA sur une grande variété d'appareils.

> XML

Le XML (Extensible Markup Language ou langage de balisage extensible) n'est pas vraiment un langage de développement. Il est principalement utilisé pour stocker des données de façon structurée. Sous Android, on l'utilise pour stocker les éléments liés à une vue, dont je vous expliquerai les détails un peu plus tard.

ENVIRONNEMENT DE DÉVELOPPEMENT (IDE)

Pour développer des applications Android, il est conseillé d'utiliser Eclipse qui est l'un des IDE les plus utilisés pour le développement en JAVA. En complément, il est nécessaire d'installer le SDK (Software Development Kit ou Kit de développement logiciel) d'Android qui contient les outils nécessaires pour créer une application. Il est également possible d'utiliser certains plugins comme le Developer Tools, qui intègre une interface graphique pour une gestion plus simple des éléments sur une vue. Si vous n'avez pas de smartphone Android à disposition, vous pouvez en simuler un. En effet, le SDK d'Android comporte un émulateur qui vous permettra d'imiter un vrai smartphone. Cet émulateur est très pratique puisqu'il permet de choisir un modèle en particulier, ou même d'en personnaliser un en spécifiant la

résolution, la version d'Android, le processeur et bien d'autres paramètres. Pour ceux qui possèdent un smartphone sous Android, il est possible de tester l'application directement sur son mobile. Une fois l'application compilée, il suffit de prendre le fichier avec l'extension « .apk » dans le dossier « workspace/VotreProjet/bin » et de le copier dans votre smartphone. Ensuite il suffit de télécharger une application (exemple : Apk Manager) qui installera votre application.

LES DIFFÉRENTES « COUCHES » D'ANDROID

> Une activité

Une activité est la composante principale pour une application Android. Elle représente l'implémentation et les interactions de vos interfaces. On pourrait comparer une activité à la page d'une application. Celle-ci contient une vue qui correspond à l'affichage des éléments ainsi que le code pour le traitement.

Une activité a plusieurs méthodes à sa disposition. Voici un résumé des méthodes qui y sont associées.

Liste des méthodes d'une activité :

- **onCreate ()** : La méthode onCreate() est appelée à la création de votre activité. C'est la première à laquelle on fait appel. Elle sert à initialiser votre activité ainsi que toutes les données nécessaires à cette dernière. Elle fournit également un bundle qui permet de retrouver votre activité dans l'état où vous l'avez laissée. onCreate() est toujours suivie par la méthode onStart().
- **onStart ()** : La méthode onStart() indique le commencement de votre activité, elle peut être suivie par un onResume() en cas de bon fonctionnement ou par un onStop() en cas de problème.
- **onRestart ()** : La méthode onRestart() est appelée après que votre activité se soit arrêtée suite à la méthode onStop(). Elle sera toujours suivie par un onStart().
- **onResume ()** : La méthode onResume() indique que l'activité se trouve au premier plan et que l'on peut interagir avec l'utilisateur. Cette méthode sera toujours suivie par un onPause().
- **onPause ()** : La méthode onPause() est appelée lors du passage d'une autre activité au premier plan. Elle permet de sauvegarder l'état de l'activité et des différents traitements effectués par l'utilisateur. Cette méthode sera suivie par un onResume() si on souhaite mettre notre activité au premier plan, ou par un onStop() si on souhaite arrêter l'activité.

- **onStop ()** : La méthode `onStop()` est appelée lorsque l'activité n'est plus visible par l'utilisateur car une autre activité est au premier plan. Elle est suivie par un `onRestart()` dans le cas où on devrait interagir de nouveau avec l'activité ou par un `onDestroy()` si on n'a plus besoin de l'activité.
- **onDestroy ()** : La méthode `onDestroy()` est l'appel final avant que l'activité ne soit détruite, elle représente la fin d'une activité. Toutes les données non sauvegardées seront perdues.

Voici une représentation du cycle de vie d'une activité : [Fig.1].

> Les services

Un service est un composant d'application qui peut effectuer des opérations en arrière plan. Il est nécessaire lorsque votre application souhaite effectuer des opérations ou des calculs en dehors de l'interaction utilisateur. Contrairement à une activité, il ne dispose pas de vue et plusieurs services peuvent être exécutés en parallèle.

Un service peut prendre deux formes :

- **Started (commencé)** : Ce service s'exécute dans des processus indépendants de votre application. Il peut tourner continuellement en arrière plan même si le composant qui l'a démarré a été détruit.
 - **Bound (lié)** : C'est un service qui s'exécute dans le même processus que votre application. Il est lié à un ou plusieurs composants, il sera exécuté aussi longtemps que les autres composants seront exécutés.
- Pour ce qui est des méthodes, on en trouve trois nouvelles, sachant que le cycle d'un service est différent de celui d'une activité.

Liste des méthodes d'un service :

- **onStartCommand()** : La méthode est appelée quand un autre composant, comme une activité, demande que le service soit démarré. Une fois cette méthode exécutée, le service démarre et peut fonctionner en arrière plan indéfiniment.
- **onBind ()** : Cette méthode est appelée lorsqu'un autre composant veut se lier avec le service.
- **onUnBind()** : La méthode est appelée lorsque tous les clients ont été déconnectés d'une interface [Fig.2].

LES VUES

Une vue peut être comparée à l'affichage d'une page sur une application. Ainsi, tous les composants visibles sur une page font partie d'une vue, que ce soit un bouton, un champ de texte ou un autre élément. Il est possible de regrouper des vues grâce à la classe `View-Group` afin de mieux structurer son application. Une vue est composée d'un layout pour la mise en page ainsi que de composants.

> Les layouts

Un layout définit la structure visuelle d'une interface utilisateur, on pourrait l'assimiler à la mise en page d'une vue. Il représente l'agencement des différents éléments graphiques d'une interface. Il est possible de créer des layouts de deux façons différentes. La première consiste à déclarer les éléments d'interface en XML car Android fournit un vocabulaire XML simple qui correspond à la vue des classes et sous-classes. La seconde possibilité, est d'utiliser les outils fournis avec Android SDK permettant de paramétrer les éléments de la vue via une interface simple d'utilisation.

Principaux Layouts :

FrameLayout : Ce layout permet d'afficher un élément dans votre interface.

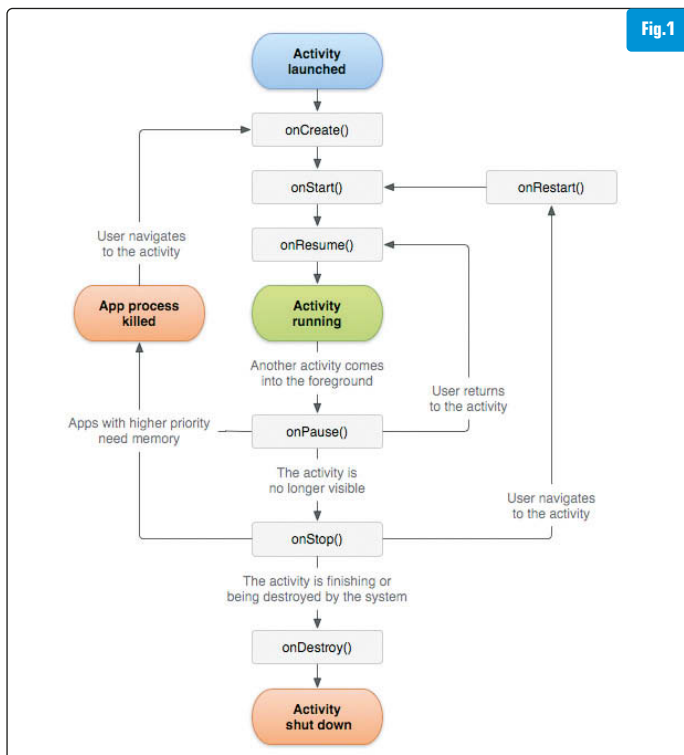
LinearLayout : Ce layout organise les différents éléments de votre interface sur une ligne ou sur une colonne.

RelativeLayout : Ce layout permet de définir la position des éléments en fonction de la position de leurs éléments parents.

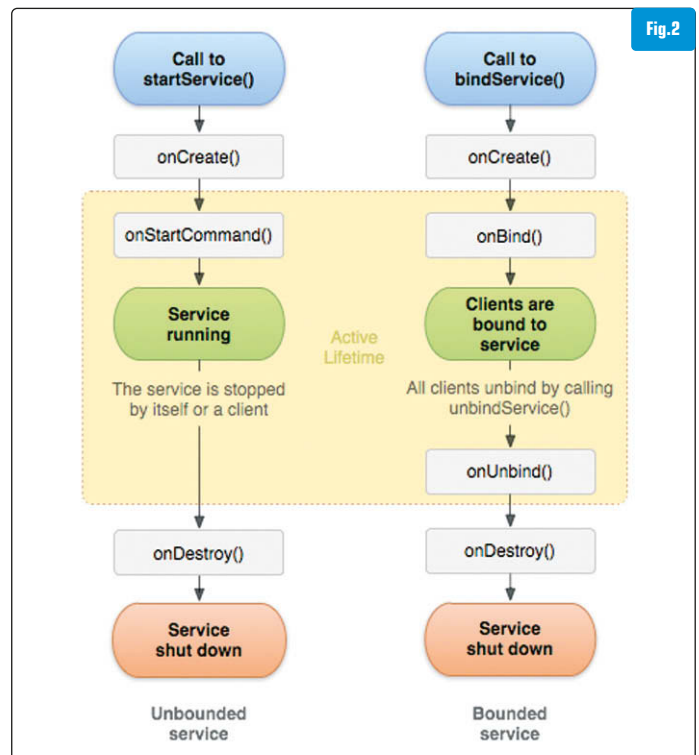
GridLayout : Ce layout permet d'organiser une vue sous forme d'un tableau composé de plusieurs lignes et colonnes

GridView : Ce layout affiche des éléments d'une à deux dimensions dans une grille.

ListView : Ce layout affiche une liste d'éléments sur une colonne scrollable.



Cycle de vie d'une activité (source developer.android)



Cycle de vie d'un service (source developer.android)

> Les composants d'une vue

Il existe divers composants à utiliser sur une vue, en voici les principaux.

Le bouton : Le bouton ou « Button » qui représente un bouton-poussoir pouvant être activé suite à une pression afin d'effectuer une action.

L'affichage de texte : L'affichage de texte que l'on nomme « TextView » permet d'afficher une chaîne de caractères dont on aura spécifié le contenu. Le texte ne pourra pas être modifié par l'utilisateur.

Le champs de texte : Le champ de texte appelé également « TextEdit » permet à l'utilisateur d'écrire une chaîne de caractères grâce au clavier virtuel d'Android.

Les images : Une image « ImageView » peut être affichée sur la vue.

Le bouton d'image : Le bouton d'image ou « ImageButton » est l'union entre « ImageView » et « Button ». Il permet donc d'exécuter une action lorsque l'on appuie sur l'image.

PLACER DES ÉLÉMENTS SUR LA VUE

Après avoir vu les principaux layouts et composants d'une vue, intéressons-nous maintenant à leur utilisation en programmation. Comme expliqué précédemment, vous pouvez éditer une vue de deux façons. Sachant que le plugin d'édition d'interface sous Eclipse est très facile d'utilisation, nous allons voir comment insérer des composants directement dans le fichier XML. Vous pouvez visualiser ci-dessous le code XML d'une vue. Bien sûr, il est fonctionnel, mais il n'affichera rien car aucun composant n'a encore été déclaré. Commençons par expliquer le code suivant :

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    tools:context=".MainActivity" >

    <!-- Divers composants -->

</RelativeLayout>
```

On peut voir que la première chose à faire est de spécifier le layout que vous désirez : ici c'est le « RelativeLayout ». Les deux premières lignes sont généralement les mêmes. La 3ème ligne correspond à la largeur du layout qui peut avoir quatre valeurs (match_parent, fill_parent, wrap_content, un nombre de pixels). La 4ème ligne définit la hauteur du layout et la 5ème est utilisée pour indexer le layout, ici il sera centré de façon horizontale. Enfin, la 6ème ligne correspond à l'activité à laquelle est associée la vue. Maintenant, passons au code d'un composant qui se rajoutera entre le contexte de la vue et le layout.

```
<Button
    android:id="@+id/buttonNextActivity"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="124dp"
    android:gravity="left|center_vertical"
    android:text="@string/NextActivity" />
```

Voici le code permettant d'insérer un bouton dans votre vue. La première chose à faire lorsque l'on crée un nouvel élément dans la vue est de lui spécifier le type d'élément que l'on veut afficher. Ici ce sera un bouton (Button) auquel on aura donné l'id « buttonNextActivity ». Les paramètres suivants l'id jusqu'à « gravity » permettent de placer le bouton. Enfin le paramètre « text » contient la chaîne de caractères « NextActivity ».

SWITCHER ENTRE DES ACTIVITÉS

Maintenant que l'on sait comment ajouter une vue et les composants qui lui sont dédiés, voyons comment faire pour changer d'activité grâce à un bouton. Tout d'abord il faut avoir créé deux activités. Pour cela, vous pouvez vous aider du code suivant :

Code de la première activité :

```
public class MainActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        final Button loginButton = (Button) findViewById(R.id.buttonNextActivity);
        loginButton.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

Code de la seconde activité :

```
public class SecondActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        final Button loginButton = (Button) findViewById(R.id.buttonPreviousActivity);
        loginButton.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent intent = new Intent(SecondActivity.this, MainActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

Comme vous pouvez le constater, le code est similaire à quelques exceptions près. Pour commencer, après avoir instancié votre classe, on utilise la méthode « onCreate() » qui va créer l'activité. Ensuite on va lier l'activité avec la vue qui correspond grâce à la méthode « setContentView ». Par exemple dans le code de la première activité, on lie notre première vue « R.layout.activity_main » avec « MainActivity » qui est la première activité. On peut voir qu'un bouton est créé et lié au bouton de la vue correspondante grâce à son id. On utilise ensuite la méthode « setOnClickListener() » qui va permettre d'exécuter une fonction lorsque l'on cliquera dessus. Cette fonction est celle qui se trouve juste après et qui se nomme « onClick ». Elle va permettre de démarrer la seconde activité. Ainsi, une fois que l'on aura appuyé sur le bouton, la nouvelle activité se lancera. Voici une représentation des deux activités dont le code a été présenté précédemment [Fig.3 et 4]. Désormais vous savez créer des activités, insérer des éléments dans une vue et gérer l'action lors d'une pression sur un bouton. Bien entendu, d'autres éléments sont encore à explorer c'est pourquoi je vous conseille de consulter le site <http://developer.android.com/> qui contient de nombreux éléments relatifs au développement sur Android.

Jonathan Lequeux

Développeur d'applications web à l'ETNA.



Fig.3

La première activité

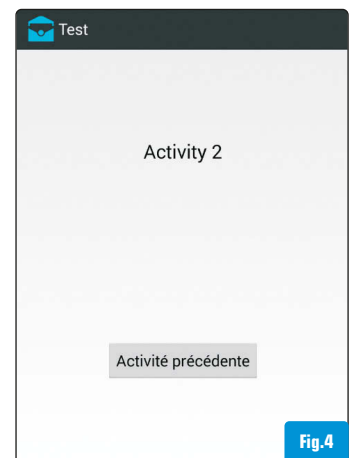


Fig.4

La seconde activité

Introduction à la sécurité avec Symfony2



Le composant sécurité du framework PHP Symfony2 est un outil puissant permettant de mettre en place une segmentation d'accès aux pages et/ou aux données très simplifiée. Il s'agit d'un portage du composant sécurité du framework Java Spring.

Les deux notions principales à retenir sont l'authentification et l'autorisation. Gardez à l'esprit que le composant sécurité n'est pas facile à appréhender dans le détail. Ceci dit, il est simplissime à utiliser ! Ainsi, je vous propose de mettre en place une limitation d'accès à certaines pages pour certains utilisateurs s'étant connectés à l'application à l'aide d'une authentification http (formulaire login/mot de passe), puis je vous donnerai quelques pistes pour approfondir le sujet.

> Le code

Notre application exemple est constituée de quatre pages :

- une page pour les utilisateurs non connectés,
- une page pour les utilisateurs connectés,
- une page pour les administrateurs connectés,
- et enfin une page de connexion.

Tout d'abord, il vous faut créer trois pages. Créez les fichiers twig suivants :

- la page pour les **utilisateurs non connectés**, dans le dossier Default, créez le fichier `index.html.twig` et collez le code suivant :

```
{% extends «::base.html.twig» %}

{% block body %}
<h1>Page d'accueil générale</h1>
{% endblock %}
```

- la page pour les **utilisateurs connectés et administrateurs**, dans le dossier Default, créez le fichier `admin.html.twig` et collez le code suivant :

```
{% extends «::base.html.twig» %}

{% block body %}
<h1> Zone administrateur</h1>
{% endblock %}
```

- la page pour les **utilisateurs connectés**, dans le dossier Default, créez le fichier `todolist.html.twig` et collez le code suivant :

```
{% extends «::base.html.twig» %}

{% block body %}
<h1>Zone utilisateur</h1>
{% endblock %}
```

Créez les routes associées comme suit (dans le fichier `routing.yml` de votre bundle) :

```
home_user:
  pattern: /welcome
  defaults:
    _controller: FrameworkBundle:Template:template
    template: MyCompanyToDoListBundle:Default:index.html.twig

admin:
  pattern: /admin
  defaults:
    _controller: FrameworkBundle:Template:template
    template: MyCompanyToDoListBundle:Default:admin.html.twig

todolist:
  pattern: /
  defaults:
    _controller: FrameworkBundle:Template:template
    template: MyCompanyToDoListBundle:Default:todolist.html.twig
```

Il est temps de sécuriser l'ensemble de nos pages. Pour cela, ouvrez le fichier `app/config/security.yml`, et écrivez le code suivant :

```
security:
  encoders:
    Symfony\Component\Security\Core\User\User: plaintext

  role_hierarchy:
    ROLE_ADMIN:    ROLE_USER

  providers:
    in_memory:
      memory:
        users:
          user: { password: userpass, roles: [ 'ROLE_USER' ] }
          admin: { password: adminpass, roles: [ 'ROLE_ADMIN' ] }

  firewalls:
    dev:
      pattern: ^/(_(profiler|wdt)|css|images|js)/
      security: false
    secured_area:
      pattern: ^/
      anonymous: ~
      form_login:
        login_path: /login
        check_path: /login_check

  access_control:
```

```
- { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
- { path: ^/welcome, roles: IS_AUTHENTICATED_ANONYMOUSLY }
- { path: ^/admin, roles: ROLE_ADMIN }
- { path: ^/, roles: ROLE_USER }
```

Et enfin, il ne nous reste plus qu'à créer notre formulaire de connexion. Pour cela, créez une classe de contrôleur SecurityController contenant le code suivant :

```
<?php

namespace MyCompany\ToDoListBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\Security\Core\SecurityContext;

class SecurityController extends Controller
{
    public function loginAction()
    {
        $request = $this->getRequest();
        $session = $request->getSession();
        if ($request->attributes->has(SecurityContext::AUTHENTICATION_ERROR)) {
            $error = $request->attributes->get(SecurityContext::AUTHENTICATION_ERROR);
        } else {
            $error = $session->get(SecurityContext::AUTHENTICATION_ERROR);
            $session->remove(SecurityContext::AUTHENTICATION_ERROR);
        }
        return $this->render('MyCompanyToDoListBundle:Security:login.html.twig', array(
            'last_username' => $session->get(SecurityContext::LAST_USERNAME),
            'error' => $error,
        ));
    }
}
```

Complétez votre fichier de routing avec les routes suivantes :

```
login_check:
    pattern: /login_check

login:
    pattern: /login
    defaults: { _controller: MyCompanyToDoListBundle:Security:login }
```

[Fig.1]

La configuration (security.yml)

Et voilà ! Parcourons maintenant le code que nous venons de décrire pour mieux le comprendre. Commençons par le fichier `security.yml` : souvenez-vous, je vous ai parlé de deux principes de bases, l'authentification et l'autorisation. C'est ce que nous construisons dans ce fichier de configuration.

L'authentification Fournisseur (provider)

Nous souhaitons restreindre l'accès à nos pages commençant par :

- « / » aux utilisateurs ayant le rôle « ROLE_USER » ou « ROLE_ADMIN »,
- « /admin » aux utilisateurs ayant le rôle « ROLE_ADMIN ».

Nous souhaitons également que les pages « /welcome » et « /login » soient accessibles aux utilisateurs ayant le rôle « IS_AUTHENTICATED_ANONYMOUSLY ». Nous devons dans un premier temps créer les utilisateurs avec ces rôles :

```
providers:
    in_memory:
        memory:
            users:
                user: { password: userpass, roles: [ 'ROLE_USER' ] }
                admin: { password: adminpass, roles: [ 'ROLE_ADMIN' ] }
```

Pour l'utilisateur ayant pour identifiant :

- « user » et le mot de passe « userpass », le rôle associé est « ROLE_USER » ;
- « admin » et le mot de passe « adminpass », le rôle associé est « ROLE_ADMIN ».

NB : Vos utilisateurs peuvent venir de n'importe quel fournisseur (une base de données, Facebook...), il vous faut simplement créer votre fournisseur (provider) à cet endroit. Consulter la documentation officielle pour apprendre à créer votre propre provider (http://symfony.com/fr/doc/current/cookbook/security/custom_provider.html).

Pare-feu (firewall)

Pour la clé « secured_area » :

```
secured_area:
    pattern: ^/
    anonymous: ~
    form_login:
        login_path: /login
        check_path: /login_check
```

Nous indiquons que :

- toutes les url commençant par « / » doivent être sécurisées ;

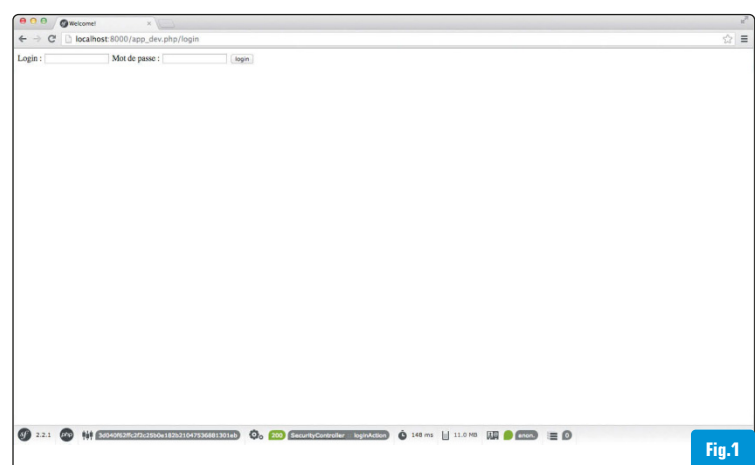


Fig.1

Formulaire de connexion

- que les utilisateurs anonymes sont autorisés (ayant le rôle « IS_AUTHENTICATED_ANONYMOUSLY ») ;
- et que notre formulaire de connexion est sur l'url « /login » (en postant sur « /login_check »).

NB : En ayant ajouté la directive « anonymous : ~ », tous les utilisateurs n'ayant pas été authentifiés en tant que « user » ou « admin », seront authentifiés en tant que « anonymous ». Ce type d'utilisateur possède le rôle « AUTHENTICATED_ANONYMOUS » par défaut (ce rôle est attribué par défaut par Symfony2).

Pour la clé « dev » :

```
dev:
  pattern: ^/(_profiler|wdt)|css|images|js/
  security: false
```

Nous indiquons que toutes les url commençant par « /_profiler », « /_wdt » (la web debug toolbar), « /css », « /images » ou « /js » ne sont pas sécurisées, donc toujours accessibles en environnement de développement.

L'autorisation

Après analyse du pare-feu, Symfony se charge de lire (de haut en bas) la liste des rôles ayant accès au type d'url (à l'aide d'une expression régulière) indiqué dans l'argument « path ». Dans notre exemple, nous indiquons que le rôle :

- ROLE_ADMIN a accès aux url commençant par « /admin » ;
- ROLE_USER a accès aux url commençant par « / ».

```
access_control:
  - { path: ^/login, roles: IS_AUTHENTICATED_ANONYMOUSLY }
  - { path: ^/welcome, roles: IS_AUTHENTICATED_ANONYMOUSLY }
  - { path: ^/admin, roles: ROLE_ADMIN }
  - { path: ^/, roles: ROLE_USER }
```

Nous avons également indiqué une hiérarchie dans les rôles avec les lignes suivantes :

```
role_hierarchy:
  ROLE_ADMIN:  ROLE_USER
```

Ainsi, le rôle admin a les mêmes droits d'accès que le rôle « ROLE_USER ». Donc concrètement, tous les utilisateurs ayant le rôle « ROLE_ADMIN » ont accès à toutes les pages.

NB : La web debug toolbar vous donne, d'une requête à l'autre, quel est l'utilisateur authentifié sur votre application.

[Fig.2] [Fig.3] et [Fig.4]

> Manipulation de l'objet User

Symfony2 vous offre une manipulation d'utilisateur grâce à la classe User située dans le dossier `vendor/symfony/symfony/src/Symfony/Component/Security/Core/User/User.php`. Il vous est possible de récupérer votre utilisateur courant grâce au conteneur de service, dans ce que nous appelons le « security context » comme suit : `$this->get('security.context')->getToken()->getUser()`. De nombreuses méthodes sont disponibles pour récupérer les informations de votre utilisateur courant, n'hésitez pas à regarder le code de cette classe.

> Manipulation de l'objet et gestion de l'utilisateur depuis une base de données

Vous admettez qu'il est plutôt rare de stocker les utilisateurs d'une application dans le fichier `security.yml` de son application. Je vous propose donc de voir comment créer un nouveau fournisseur (provider) très simple. Dans un premier temps, il vous faut créer votre propre classe « User » : il faut que votre classe implémente l'interface « UserInterface » (je vous invite à jeter un coup d'œil au fichier `UserInterface.php` situé dans le dossier `vendor/symfony/symfony/src/Symfony/Component/Security/Core/User`).

Votre classe doit obligatoirement implémenter les méthodes suivantes :

- getUsername()
- getPassword()
- getSalt()
- getRoles()
- et eraseCredentials().

Créez ensuite votre nouveau provider dans le fichier `security.yml` :

```
security:
  providers:
    main_provider:
      entity:
        class: MyCompany\TodoListBundle\Entity\User
        property: username
```

> Aller plus loin

Le sujet de la sécurité avec Symfony2 est très vaste ! Cet article n'a pas vocation à tout traiter. Ceci dit, voici quelques tuyaux pour approfondir le sujet de manière méthodique.

> Limiter la visibilité de ses ressources

L'un des besoins les plus récurrents est de limiter la visibilité des ressources que vous présentez dans votre application. Il est possible

Fig.2 Authentifié en tant qu' « anonymous »




Fig.3 Authentifié en tant que «user»





Fig.4 Authentifié en tant que «admin»



de limiter cette visibilité depuis votre contrôleur grâce au test simple sur le ou les rôles de votre utilisateur :

```
if ($this->get('security.context')->isGranted('ROLE_ADMIN')) {
    // Code uniquement nécessaire aux utilisateurs ayant le rôle admin
}
```

Pour une limitation encore plus fine, il vous est également possible de tester le ou les rôles de votre utilisateur courant, au sein même de vos templates :

```
{% if is_granted('ROLE_ADMIN') %}
    Contenu uniquement visible par les utilisateurs ayant le rôle admin.
{% endif %}
```

> Forcer le HTTPS sur certaines URL

Dans le cas où il est nécessaire d'utiliser le protocole HTTPS sur une partie de votre application, sachez qu'il est également possible de forcer l'accès à ces parties à certains utilisateurs via leur rôle.

Prenons un exemple : sur l'ensemble du tunnel de commande de votre site e-commerce, vous devez utiliser le protocole HTTPS.

Dans notre exemple, toutes les urls de votre tunnel de commande commencent par « /purchase ». Nous souhaitons également limiter l'accès à ces pages uniquement aux utilisateurs authentifiés avec le rôle « ROLE_USER ». Ainsi, il vous faut donc ajouter la directive suivante dans votre fichier `security.yml` :

```
access_control:
    path: ^/purchase
    roles: ROLE_USER
    requires_channel: https
```

■ **NB** : Il est possible d'ajouter autant de directives que nécessaire.

> Se documenter et apprendre

Je vous invite vivement à creuser le sujet en regardant tout simplement le code du composant sécurité sur Github :

<https://github.com/symfony/Security>.

Johannes Schmitt, un des nombreux contributeurs au composant sécurité a également développé le `JMSecurityExtraBundle`, ajoutant quelques fonctionnalités intéressantes au dessus du composant Sécurité de Symfony2 (comme d'utiliser les annotations pour la configuration des autorisations). Vous retrouverez tout ce dont vous aurez besoin sur le site officiel du bundle :

<http://jmsyst.com/bundles/JMSecurityExtraBundle#overview>.

Les conférences sont aussi une source inépuisable de conseils et de savoir. Je vous invite à regarder les deux conférences suivantes :

- « Love and loss » par Kris Wallsmith (en anglais) <http://bit.ly/1as8cky> ;
- « Bien comprendre le cœur du composant sécurité » par Joseph Rouff (développeur chez SensioLabs) <http://bit.ly/14BaDJO>.

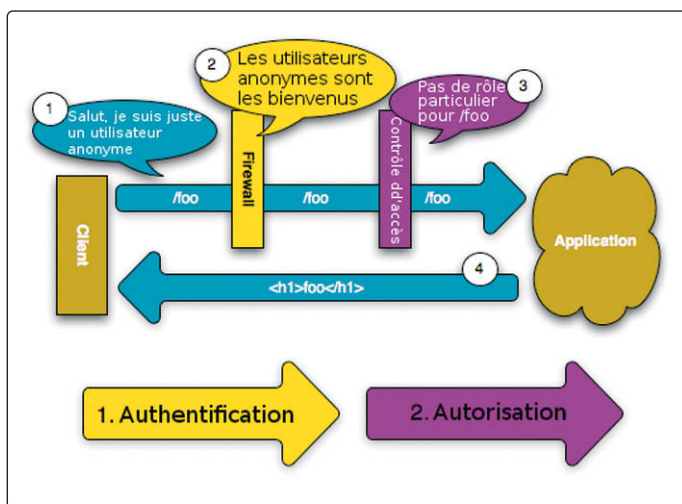
Pensez également à suivre la mailing list officielle Symfony2 pour un support gratuit, mais aussi aider d'autres membres de la communauté ! <https://groups.google.com/forum/#!forum/symfony2>

Et enfin, encore et toujours, la documentation officielle de Symfony2 sur le site <http://symfony.com>, section documentation puis **book** (<http://symfony.com/doc/current/book/security.html>), **cookbook** (<http://symfony.com/doc/current/cookbook/security/index.html>) ou encore **component** (<http://symfony.com/doc/current/components/security/index.html>).

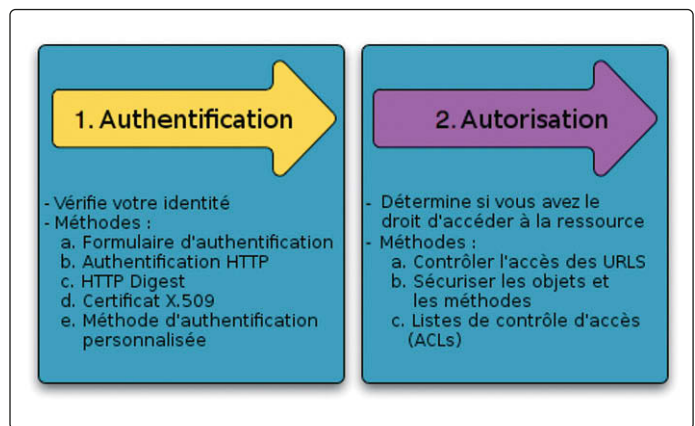
Il s'agit du dernier article de la série de trois sur l'introduction à Symfony2. Bien évidemment, nous n'avons pas couvert tous les (très nombreux) sujets sur le framework. Néanmoins, j'espère vous avoir donné les premières clés pour une première utilisation !

Sarah Khalil - Chef de Projet technique SensioLabs

sarah.khalil@sensiolabs.com



Accès sécurisé d'un utilisateur anonyme.



Authentification et autorisation.



ABONNEMENT PDF

30 € par an soit 2,73 € le numéro

www.programmez.com

Résoudre les problèmes de **OutOfMemoryError** en Java

*Quel développeur Java n'a jamais vu un programme planter avec une erreur de type **OutOfMemoryError** ? Cet article se propose de faire le tour des principales erreurs Java de ce type, en proposant les solutions pour y remédier.*

Bien qu'exécutés au sein d'une machine virtuelle, les programmes Java ne sont pas à l'abri des fuites mémoires. Au contraire, les applications desktop, et plus encore les applications web s'exécutant au sein de serveurs d'applications, produisent souvent des erreurs **OutOfMemoryError**.

Leur apparition est le résultat d'un problème mémoire trop important provoquant son arrêt ainsi que celui de la JVM. Avant d'en arriver là, des fuites mémoires peuvent déjà être présentes au sein d'un programme provoquant des ralentissements.

Les solutions que nous allons détailler ici pourront également s'appliquer en prévention.

> **OutOfMemoryError : Java heap space**

Lorsqu'elle survient, cette erreur est accompagnée de la stack trace suivante :

```
java.lang.OutOfMemoryError: PermGen space
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClassCond(ClassLoader.java:632)
    ...
```

Classique, cette erreur indique que la JVM, après le passage du Garbage Collector, n'a plus de place pour allouer un nouvel objet dans la pile (heap). Elle n'implique donc pas toujours une fuite mémoire et peut être résolue simplement en allouant plus de mémoire à la pile. Par défaut, sa taille maximum est positionnée à 64Mb. Pour l'augmenter, il faut utiliser l'option `-Xmx` au lancement de la JVM. On peut également augmenter sa taille au démarrage avec `-Xms` :

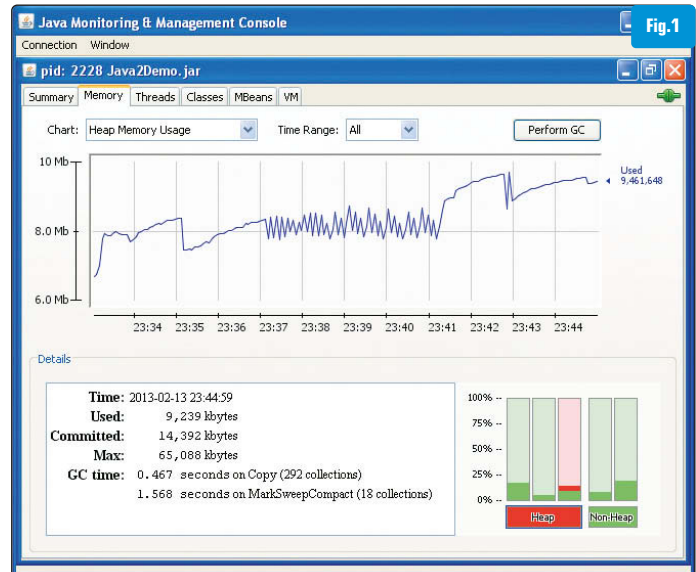
```
java -Xms256m -Xmx512m ...
```

Rapide, cette solution n'est pas forcément la meilleure si le programme incriminé a une durée de vie longue. Dans un tel cas, il s'agit bien d'une fuite mémoire et l'augmentation de la taille allouée ne fera que repousser le plantage.

A ce moment-là, il faut s'intéresser aux vrais symptômes de cette erreur. Pour ce faire, la première approche consiste à utiliser un profiler au sein d'un IDE tel que NetBeans permettant d'inspecter les allocations mémoire et de découvrir les zones où se situent les fuites. Une autre approche est de s'appuyer sur des outils fournis en standard avec le JDK.

La première étape étant d'obtenir un dump mémoire du heap du programme en cours d'exécution via `jmap` puis d'utiliser `jhat` pour l'analyser finement.

On notera également que l'outil graphique `jconsole`, plus récent, permet de directement monitorer la mémoire d'une application [Fig.1].



Monitoring avec JConsole

> **OutOfMemoryError : PermGen space**

Autre classique, l'erreur **PermGen space** concerne la seconde grande zone mémoire allouée par la JVM : la Permanent Generation. Répondant au surnom de "PermGen", elle contient notamment toutes les méthodes des classes (incluant le bytecode), les noms des classes, un pool de constantes, les tableaux d'objets associés à des classes, les objets internes créés par la JVM et des informations utilisées à des fins d'optimisation (JIT). Dans le cas d'une application web, la compilation des JSPs crée un certain nombre d'objets pouvant provoquer une erreur **PermGen space**. Le réglage par défaut allouant 64Mb de taille maximale peut alors s'avérer insuffisant. Une solution consiste à allouer plus de mémoire à la PermGen dès son démarrage et en taille maximale :

```
java -XX:PermSize=96m -XX:MaxPermSize=128m ...
```

Pour des applications au long cours, le problème provient en revanche bel et bien d'une fuite mémoire. Le phénomène est assez simple à comprendre. Lors du chargement de l'application web par le serveur d'applications, les classes de l'application sont chargées dans un `ClassLoader` par le serveur. Les définitions des méthodes et classes étant alors stockées dans la PermGen de la JVM du serveur. Lors d'un redémarrage serveur, si des références à ces objets existent encore, l'espace mémoire occupé n'est pas libéré, constituant une fuite qui provoquera un "**OutOfMemoryError : PermGen space**".

au bout de quelques redémarrages. Les causes les plus communes étant :

- Le chargement d'un driver JDBC dans le ClassLoader, qui devient référencé au niveau serveur par le DriverManager et n'est donc pas désalloué
- Une bibliothèque tierce, telle que Hibernate, référencée dans un cache des classes de l'application
- L'utilisation d'un thread, géré par le pool de threads du conteneur, qui continue à vivre une fois l'application stoppée

Toutes ces causes provoquent des fuites mémoire ne pouvant être résolues durablement via une augmentation de la taille de la PermGen. Les solutions se trouvent au niveau applicatif, avec des mesures pour libérer les références aux objets et ainsi permettre un travail efficace du Garbage Collector. Par exemple, on utilisera un ServletContextListener pour décharger le driver lors du stop de l'application web. Pour localiser au mieux les zones responsables de ces fuites, il faut s'appuyer sur l'outil jmap lancé avec l'option -permgen qui fournit des statistiques complètes sur les objets présents au sein de la PermGen. Il est bon de noter que jconsole est également une solution permettant d'obtenir un rendu graphique du contenu de la PermGen.

> OutOfMemoryError : GC overhead limit exceeded

Apparue avec le JDK 1.6, cette erreur est un peu différente des précédentes. Ici, la JVM ne dit pas qu'elle manque d'espace mémoire sur le heap ou la PermGen, mais signale que l'application se trouve à la limite des possibilités mémoire. De fait, durant son exécution plus de 98% du temps est consacré au Garbage Collector pour seulement 2% à l'application. Cette erreur permet d'éviter des écueils importants et il est donc préférable de ne pas la désactiver à l'aide de l'option -XX:UseGCOverheadLimit. La solution se situe plutôt au niveau applicatif avec la nécessité d'optimiser l'algorithme responsable de ce surcoût de consommation mémoire.

> OutOfMemoryError : Requested array size exceeds VM limit

Liée à un problème d'espace mémoire de la heap, cette erreur est plus détaillée puisqu'elle cible une demande d'allocation dynamique d'un tableau. Cette demande dont la taille dépasse la limite fixée par la JVM qui peut se présenter via la stack trace suivante :

```
java.lang.OutOfMemoryError: Requested array size exceeds VM limit
    at java.util.Arrays.copyOf(Unknown Source)
    at java.lang.AbstractStringBuilder.expandCapacity(Unknown Source)
    ...
```

Ce type d'erreur dénote bien souvent un problème de configuration au niveau applicatif avec une erreur dans le calcul fixant la taille du tableau à allouer. La solution est donc de vérifier si la taille du tableau qu'on tente de créer est correcte, et, le cas échéant, d'augmenter la taille de la pile via l'option -Xmx.

> OutOfMemoryError : request <size> bytes for <reason>. Out of swap space?

Cette erreur est remontée par la JVM quand elle n'arrive pas à allouer de la mémoire native sur l'OS sous-jacent. Il s'agit d'une erreur totalement différente des OutOfMemoryError liée à la pile ou la PermGen. Elle se produit généralement lorsque l'OS atteint ses limites. Le réflexe d'ajouter de la mémoire au démarrage de la JVM n'a donc aucun impact.

La solution consiste à diagnostiquer la mémoire du côté du système sous-jacent; il pourrait être configuré avec un espace de swap insuffisant, ou bien, un autre processus sur le système consomme toutes les ressources mémoire. En dernier recours après ces vérifications, on peut alors s'interroger sur son application en cherchant si elle provoque des fuites de mémoire natives en allouant continuellement de la mémoire sans jamais la libérer.

> OutOfMemoryError : <reason> <stack trace> (Native method)

L'utilisation de routines C, C++ en Java via JNI peut conduire à des fuites mémoires qui se manifestent par cette erreur. Si les fuites mémoire sont détectées par la JVM au niveau du code natif, elle renvoie ce type d'erreur précisant que l'allocation a échoué du côté du code C/C++. Il faut donc aller analyser directement l'exécution de ce dernier ou bien demander de l'aide à nos amis développeurs C, C++.

> Pour finir

Connues de tous les développeurs Java, les erreurs OutOfMemoryError ne doivent pas être prises à la légère en laissant dissimuler les symptômes par l'ajout de mémoire à la heap ou à la PermGen. En effet, elles résultent très souvent de réels problèmes causant des fuites mémoire. Il faut donc les combler afin de développer des applications plus robustes.

En outre, les outils permettant d'analyser et de monitorer la mémoire des applications Java sont nombreux, et bon nombre sont fournis en standard dans le JDK.

En s'appuyant sur ces outils, les OutOfMemoryError deviendront très rapidement un lointain souvenir.

Sylvain Saurel – Ingénieur d'Etudes Java / Java EE
sylvain.saurel@gmail.com

Abonnement : Programmez, 17, Route des Boulangers, 78926 Yvelines Cedex 9 - Tél. : 01 55 56 70 55 - abonnements.programmez@groupe-gli.com - Fax : 01 55 56 70 91 - du lundi au jeudi de 9h30 à 12h30 et de 13h30 à 17h00, le vendredi de 9h00 à 12h00 et de 14h00 à 16h30.
Tarifs abonnement (magazine seul) : 1 an - 11 numéros France métropolitaine : 49 € - Etudiant : 39 € - CEE et Suisse : 55,82 € - Algérie, Maroc, Tunisie : 59,89 € Canada : 68,36 € - Tom : 83,65 € - Dom : 66,82 € - Autres pays : nous consulter.
PDF : 30 € (Monde Entier) souscription exclusivement sur www.programmez.com



Une publication Nefer-IT
7 avenue Roger Chambonnet
91220 Brétigny sur Orge
diffusion@programmez.com
Tél. : 01 60 85 39 96

Directeur de la publication & rédacteur en chef : François Tonic
Ont collaboré à ce numéro : J-B Boisseau, S. Saurel
Secrétaire de rédaction : Olivier Pavie
Experts : C. Villeneuve, C. Chervy, Greg, A. Villenave, V. Fabing, K. Alnires, Y. Vigara, S. Bataouche, V. Chartier, S. Cailliau, D. Lefranc, B. Boucard, J-B Mercy, J. Corioland, T. Leriche-Dessirier, G. Beaumont, J. Lequeux, S. Khalil
Crédits couverture : Frankfurt on the Main: Gerechtigkeitsbrunnen (Fountain of Justice), detail of the Lady Just-

ce © Mylius (image sous licence CC), Google, Oracle (Duke), Samsung, Intel - **Maquette** : Pierre Sandré

Publicité : Régie publicitaire, K-Now sarl. Pour la publicité : Tél. : 01 41 77 16 03 - diff@programmez.com.

Imprimeur : S.A. Corelio Nevada Printing, 30 allée de la recherche, 1070 Bruxelles, Belgique.

Ce numéro comporte : 1 encart jeté Component Source sur une partie du tirage, 1 encart jeté 1&1 sur la totalité du tirage - Imprimé en Belgique / Printed in Belgium

Marketing et promotion des ventes : Agence BOCONSEIL - Analyse Media Etude
Directeur : Otto BORSCHA oborsch@boconseilame.fr
Responsable titre : Terry MATTARD

Contacts

Rédacteur en chef : ftonic@programmez.com
Rédaction : redaction@programmez.com
Webmaster : webmaster@programmez.com
Publicité : diff@programmez.com
Evenements / agenda : redaction@programmez.com

Dépôt légal : à parution - Commission paritaire : en cours - ISSN : 1627-0908

© NEFER-IT / Programmez, octobre 2013
Toute reproduction intégrale ou partielle est interdite sans accord des auteurs et du directeur de la publication.

Les outils des Décideurs Informatiques

Vous avez besoin d'info
sur des sujets
d'administration,
de sécurité, de progiciel,
de projets ?

Accédez directement
à l'information ciblée.

Cas clients
Actu triée par secteur
Avis d'Experts



Actus / Evénements | Newsletter | Vidéos

www.solutions-logiciels.com

☐ **OUI, je m'abonne** (écrire en lettres capitales)

Envoyer par la poste à : Solutions Logiciels, service Diffusion, GLIE - 17 chemin des Boulangers 78926 Yvelines cedex 9 - ou par fax : 01 55 56 70 20
1 an : 50€ au lieu de 60€, prix au numéro (Tarif France métropolitaine) - Autres destinations : CEE et Suisse : 60€ - Algérie, Maroc, Tunisie : 65€ , Canada : 80€ - Dom : 75€ Tom : 100€
10 numéros par an.

☐ M. ☐ Mme ☐ Mlle Société

Titre : Fonction : ☐ Directeur informatique ☐ Responsable informatique ☐ Chef de projet ☐ Admin ☐ Autre

NOM Prénom

N° rue

Complément

Code postal : [] [] [] [] Ville

Adresse mail

☐ Je joins mon règlement par chèque à l'ordre de SOLUTIONS LOGICIELS ☐ Je souhaite régler à réception de facture



WINDEV®

COMMANDEZ WINDEV 19

ou WINDEV Mobile 19 ou WEBDEV 19

CHEZ PC SOFT ET RECEVEZ UN MATÉRIEL AU CHOIX POUR «1 EURO DE PLUS»

Choisissez votre matériel :



**(x2) Nouveau
smartphone
Samsung Galaxy S4**

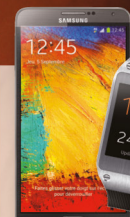
Ou choisissez la
version «S4 mini» ou
«S4 Activ»
Configurations sur
www.pcsoft.fr



**(x2) Nouvelle
tablette
Samsung
Galaxy Tab 3 10p**
Ou choisissez la
version 7 ou 8 pouces
Configurations sur
www.pcsoft.fr



**DELL : PC Portable,
PC Intégré
ou Station
de Travail**
4 modèles au
choix
Configurations sur
www.pcsoft.fr



**Samsung
Galaxy Gear +
Smartphone
Samsung
Galaxy Note 3**

Configurations sur
www.pcsoft.fr

Pour bénéficier de cette offre exceptionnelle, il suffit de commander WINDEV Mobile 19 (ou WINDEV 19, ou WEBDEV 19) chez PC SOFT au tarif catalogue avant le 20 décembre 2013: pour 1 Euro de plus, vous recevrez alors le ou les magnifiques matériel(s) que vous aurez choisis. Offre réservée aux sociétés, administrations, mairies, GIE et professions libérales, en France métropolitaine. **L'offre s'applique sur le tarif catalogue uniquement.** Voir tous les détails et des vidéos sur : www.pcsoft.fr ou appelez-nous.
Le Logiciel et le matériel peuvent être acquis séparément. Tarif du logiciel au prix catalogue de 1.650 Euros HT (1.973,40 TTC). Merci de vous connecter au site www.pcsoft.fr pour consulter la liste des prix des matériels et les dates de disponibilité. Tarifs modifiables sans préavis.

Descriptif technique
complet des matériels
sur www.pcsoft.fr

Tél province: **04.67.032.032**
Tél Paris: **01.48.01.48.88**



Fournisseur Officiel de la Préparation Olympique

www.pcsoft.fr

OPÉRATION
**1 POUR
EURO
DE PLUS**

JUSQU'AU 20 DÉCEMBRE

138 CM



**Télévision SAMSUNG 138 cm
LED 3D Full HD Réf UE55F6400**
1.920 x 1.080. 200Hz DLNA Wifi 3xUSB
4xHDMI 2xLunettes 3D